

31963  
2ej. /



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES**

**I Z T A C A L A**

**"PAQUETE COMPUTACIONAL E INTERFACE PARA EL CONTROL,  
REGISTRO, ALMACENAMIENTO Y ANALISIS DE EVENTOS  
EN PSICOLOGIA EXPERIMENTAL"**

**T E S I S**

**MAESTRIA EN PSICOLOGIA**

**RUBEN PATRICIO CHAVEZ INOSTROZA**

**TESIS CON  
FALSA FE CRONOL**

**MEXICO, D. F.**

**1988**



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# I N D I C E

Páginas

## PRESENTACION .

## INTRODUCCION

Antecedentes	I.1
Objetivos	I.4
Justificación	I.5

## CAPITULO UNO: NOCIONES BASICAS

El funcionamiento eléctrico de los aparatos del A.E.C.	1.2
a) focos	1.3
b) relevadores	1.3
c) interruptores	1.4
Arquitectura básica de una microcomputadora.	1.6

## CAPITULO DOS: DESCRIPCION DE LA INTERFACE

Arquitectura de la Commodore 64	2.2
Descripción del CIA 6526 #1	2.5
Descripción del CIA 6526 #2	2.8
Requerimientos de la interface	2.10
Configuración de la interface	2.13
Las salidas de la interface	2.14
a) El multiplexador	2.14
b) Las memorias	2.16
c) Los transistores	2.17
Las entradas de la interface	2.19
Conexiones de la interface	2.19
a) Alimentación de la interface	2.19
b) Conexión de las salidas	2.20
c) Conexión de las entradas	2.21
d) Conexión a la C-64	2.22
Prueba de la interface	2.23

## CAPITULO TRES: PROGRAMACION DE LA INTERFACE

Programación de las salidas de la interface	3.2
1)Habilitación de líneas de salida	3.4
2)Direccionamiento de los datos a los puertos	3.5
3)Activación de líneas	3.6
Detección y registro de respuestas	3.9
1)Inhibición de la lectura del teclado	3.9
2)Lectura de respuestas	3.10

3) Identificación de líneas activadas	3.11
4) Registro de respuestas	3.14

#### CAPITULO CUATRO: ALGORITMOS DE PROGRAMACION

a) Rutina de reforzamiento	4.1
b) Rutina de grabado de datos	4.3
c) Manejo del tiempo	4.4
d) Manejo de probabilidades	4.7
e) Programas de razón	4.8
f) Programas de intervalo	4.12
g) Captura de parametros por pantalla	4.15
h) Análisis de datos	4.16
i) Sugerencias para elaborar un programa	4.17

#### CAPITULO QUINTO: REPORTE EXPERIMENTAL

Introducción	5.1
Método	5.2
a) Sujetos	5.2
b) Aparatos	5.2
c) Procedimiento	5.4
c) Análisis	5.4

#### APENDICES

A.- Figuras
B.- Tablas
C.- Rutinas
D.- Programas

#### BIBLIOGRAFIA

## P R E S E N T A C I O N

En la introducción de este trabajo se encontrarán los antecedentes y motivos que originaron el desarrollo de esta tesis.

En el capítulo 1 se presentan algunos conceptos básicos que serán de utilidad para comprender la interacción entre una computadora y los dispositivos que usualmente se encuentran en el ámbito de trabajo del Análisis Experimental de la Conducta.

En el capítulo 2 se describen las características y especificaciones de la interface que hace posible la comunicación entre la computadora y uno o más espacios experimentales.

En el capítulo 3 se aporta la información necesaria requerida para programar la presentación de estímulos y para detectar y registrar las respuestas ocurridas en uno o más espacios experimentales.

En el capítulo 4 se muestran diversas aplicaciones para el control, registro y análisis de eventos comunes dentro del paradigma operante por medio de la microcomputadora C-64 y la interface desarrollada. También se presentan algunas

sugerencias para elaborar los programas computacionales de acuerdo con las necesidades del usuario.

En el capítulo 5 se reportan los datos de un experimento cuya programación, control, registro y análisis se realizaron con el equipo desarrollado y descrito en esta tesis.

Finalmente, un apéndice especial contiene varias rutinas y programas computacionales elaborados para dar cuenta de algunos procedimientos propios del Análisis Experimental de la Conducta.

## I N T R O D U C C I O N

### ANTECEDENTES

El Análisis Experimental de la Conducta ha sustentado su desarrollo en la información obtenida en los laboratorios de psicología. Ello ha implicado el diseñar y adaptar diversos tipos de instrumentos y aparatos para el control, registro y almacenamiento de los eventos experimentales. La instrumentación en la investigación conductual ha ido evolucionando desde un control manual a un control efectuado por aparatos electromecánicos y de estado sólido.

En los años setenta hubo una aplicación de las minicomputadoras al campo de la instrumentación en psicología experimental y a principios de los ochenta se ha observado la utilización de microprocesadores para tales fines.

Un avance significativo en la instrumentación fue la aparición del sistema SKED, desarrollado por Snapper (1974), el que permite programar y controlar prácticamente cualquier tipo de experimento en psicología en la forma de una serie de "estados" (Snapper, 1984). Este sistema fue diseñado para ser usado en minicomputadoras relativamente costosas, lo que dificulta su empleo generalizado especialmente en países de la América Latina.

En años recientes este sistema ha sufrido varias

adaptaciones para hacerlo compatible con microcomputadoras personales tales como la IBM PC, Hewlett Packard 150, o la DEC 350. Sin embargo a pesar de las múltiples ventajas que ofrece este sistema (fácil programación, control simultáneo de varias cajas experimentales), su costo sumado al de las microcomputadoras hacen de él algo difícil de obtener en el ámbito de los países latinoamericanos.

Una alternativa viable del punto de vista de factibilidad económica pareció ser a finales de los setenta y principio de los ochenta, el surgimiento de microprocesadores de 8 bits (8080, 8085, Z-80, 6502) adaptados para su utilización en laboratorios de psicología (Butler, 1980; Perera, 1981; Rayfield, 1982). Sin embargo, a pesar del bajo costo comercial, su uso no se generalizó dada la dificultad de programación de estos microprocesadores no provisto de lenguajes de uso común tales como Fortran IV, Pascal, Basic, etc.

Otro inconveniente asociado con el uso de los microprocesadores en los laboratorios de psicología ha sido la naturaleza de la arquitectura de estos sistemas que requieren el crear un sistema mínimo para su aplicación. Este sistema mínimo implica el diseño de complejos circuitos electrónicos para interactuar con el espacio experimental y de un largo y complejo proceso de programación en lenguaje de máquina.

Las características del desarrollo de la instrumentación en psicología experimental, especialmente su

velocidad creciente de cambio, ha contribuido a que en los pocos laboratorios de psicología existentes en Latino America se observen algunas de las siguientes paradojas:

1) Existencia de equipos electrónicos (minicomputadoras y microcomputadoras) no utilizados o subutilizados (como procesadores de texto).

2) Imposibilidad de utilizar equipo electrónico de alto nivel tecnológico por falta de accesorios mínimos (interfaces) o equipo periférico complementario de costo relativamente bajo.

3) Dificultades serias para utilizar los equipos electrónicos existentes, puesto que requieren de operadores con conocimientos avanzados en programación de lenguajes de bajo nivel (lenguajes de máquina o de ensamblador) o de especialistas en diseños de circuitos electrónicos que hagan posible la interacción de una computadora con un espacio experimental.

4) Utilización preferente de equipos electromecánicos y de estado sólido cuyo costo en programación es alto y la pérdida de datos relevantes es cuantiosa.

Considerando lo descrito anteriormente se presenta el siguiente trabajo dirigido a maximizar la utilización de la capacidad instalada de los laboratorios de psicología mediante el diseño de una interface multipropósito y de un paquete de programación que le permita al usuario definir los parámetros de su experimento sin necesidad de conocimientos avanzados en programación computacional.

## OBJETIVOS:

El objetivo general del proyecto es desarrollar un paquete computacional para la utilización de microcomputadoras de bajo costo comercial, y de fácil acceso en el mercado de los países latinoamericanos, en la programación, control, registro y análisis de eventos ocurridos en espacios experimentales diversos. Para el logro de este objetivo es preciso:

1.- Diseñar y construir los circuitos electrónicos de una interface multipropósito que permita la interacción de la microcomputadora con distintos tipos de dispositivos ubicados en espacios experimentales. Eventualmente, esta interface podría controlar dos o más espacios experimentales simultáneamente.

2.- Elaborar un sistema de rutinas computacionales de aquellos procedimientos usualmente utilizados en el Análisis Experimental de la Conducta de tal modo que le permita al usuario construir su propio programa encadenando aquellas rutinas apropiadas a su procedimiento experimental, efectuando las adaptaciones necesarias sin necesidad de conocimientos avanzados en programación. Adicionalmente, el sistema permitirá el registro de una diversidad grande de variables, su almacenamiento y posterior análisis. Se considera la posibilidad de que los archivos de datos de los

experimentos sean compatibles con otros programas de análisis computacional ya disponibles en el mercado.

3.- Elaborar un manual que comprenda las especificaciones técnicas de la interface y sus conexiones con la microcomputadora, con los aparatos eléctricos y electromecánicos a utilizar en el experimento, y una guía para la utilización de los diversos subprogramas y rutinas disponibles en el paquete computacional.

4.- Adicionalmente, la integración de los tres objetivos anteriores se demuestra en el reporte de un estudio propio del Análisis Experimental de la Conducta.

#### JUSTIFICACION:

Si se considera que la investigación en ciencia básica en psicología está tendiendo a una mayor sofisticación tanto en el diseño de los experimentos como en el tipo de datos a registrar, no es raro observar el número cada vez más creciente de equipos electrónicos de alto nivel tecnológico usados en la programación y control de los experimentos en psicología (5 de los 9 experimentos reportados en el Journal of the Experimental Analysis of Behavior de Enero de 1985 . utilizaron mini o microcomputadoras para fines de programación, control, registro y análisis de los eventos

experimentales). Otras aplicaciones de la tecnología computacional se observan cada vez con mayor frecuencia en el área de la psicología aplicada, tales como: investigación en modificación de conducta y evaluación conductual (Lang, 1980), diagnóstico psiquiátrico (Greist y Klein, 1980), terapias cognitivas basadas en computadoras (Selmi, Klein, Greist, Johnson y Harris (1982).

Evaluando la tendencia creciente hacia un mayor empleo de las computadoras en la investigación básica y aplicada en psicología parece necesario y urgente el desarrollar una tecnología propia que nos permita acceder a las ventajas metodológicas, técnicas y económicas que significa la inclusión de las microcomputadoras en los laboratorios de psicología experimental.

A continuación se exponen las razones metodológicas, técnicas y económicas que justifican la realización del presente trabajo.

1.- Razones metodológicas: La investigación conductual requiere cada vez mas de un mayor número de respuestas a registrar en tiempo real, controlar la presentación de estímulos aleatorizada o sujeta a un nivel predeterminado de probabilidad o a ocurrir en determinado momento o presentación breve. Asimismo, los diseños experimentales se han ido complejizando y las exigencias de un riguroso control y registro de las variables ha ido en aumento. Las tareas a estudiar en un espacio experimental cada vez son

más variadas y el tipo de análisis de los datos que se efectúan requieren de una gran transparencia de los mismos. Esto es, cada vez es más deseable contar con los datos del tipo de evento ocurrido y el tiempo real en que este acaeció. Y estos datos deben estar almacenados de tal manera que en todo momento sea posible volver a ellos para analizarlos de acuerdo a distintos procedimientos. Además, siempre es deseable poder efectuar estudios de sujetos en forma simultánea a objeto de contrarrestar efectos producidos por el simple paso del tiempo. Skinner (1966, 1974) ha sido enfático en destacar el rol desempeñado por la tecnología en el área de la instrumentación la que ha permitido avances notables en el desarrollo teórico de la psicología.

2.- Razones técnicas: El tipo de argumentos anteriores conduce a buscar el auxilio de las computadoras para la ejecución de experimentos propios del estado de avance actual de la ciencia psicológica. Nuestra realidad económica y de dependencia tecnológica nos impide el fácil acceso a minicomputadoras y nos dificulta la fácil utilización de microprocesadores que requieren la construcción de sistemas complementarios para su empleo en un tipo de experimento particular. Esto hace que nuestra búsqueda se oriente hacia microcomputadoras de bajo costo comercial, de fácil acceso en los mercados nacionales de Latino América y que ya cuentan con un sistema operativo, un lenguaje de alto nivel,

y la posibilidad de acceder a un lenguaje de bajo nivel (de máquina o ensamblador) para operar y controlar sus modos de comunicación con el exterior. Las posibilidades de programación, de capacidad de memoria, y de utilización de un reloj con una resolución de 1/60 de segundo y la cantidad de información disponible sobre las características de la máquina y de programación hacen de la microcomputadora Commodore 64 un elemento adecuado para satisfacer las demandas metodológicas de la investigación conductual. Sin embargo, es necesario diseñar y construir una interface que permita utilizar el microprocesador 6510 de la Commodore 64 para interactuar con espacios experimentales diversos mediante la adecuada programación de su microprocesador. Esto último es objeto del presente trabajo.

3.- Razones económicas: El bajo costo comercial de la microcomputadora Commodore 64 y de sus equipos periféricos (unidad de disco o casete, monitor, impresora) hacen de este equipo algo relativamente fácil de obtener para cualquier laboratorio de psicología. Además, el costo de la interface multipropósito a desarrollarse no sobrepasa los 150 dólares, lo que significa que un equipo computacional completo apto para programar, controlar, registrar y analizar una gran variedad de eventos experimentales no sobrepasa los 1,000 dólares. Si a esto se suma el ahorro de los tiempos de programación y la independencia de personal especializado para dichos fines, aparece como altamente ventajoso del

punto de vista económico el desarrollar una tecnología orientada al uso de este tipo de microcomputadoras en los laboratorios de psicología.

## C A P I T U L O   U N O

### NOCIONES BASICAS

En el capítulo introductorio se mencionó como uno de los principales objetivos de esta tesis el controlar los eventos ocurridos en espacios experimentales diversos. El control de estos eventos, hablando en términos propios del Análisis Experimental de la Conducta, significa detectar y registrar la ocurrencia de alguna respuesta (definida por la activación de un interruptor eléctrico) la que de acuerdo a ciertas condiciones experimentales estipuladas con anterioridad determinará el accionar de aparatos eléctricos o electromecánicos para la presentación de estímulos. Por tanto, el objetivo a lograr implica por una parte la activación de aparatos eléctricos como consecuencia de la detección de la operación de uno o más interruptores eléctricos. Por otra parte es necesario controlar el procedimiento experimental a observar en el transcurso de una sesión y el posterior almacenamiento y análisis de los datos generados en la sesión experimental.

El primer aspecto del objetivo planteado dice relación con el manejo y control de aparatos eléctricos y electromecánicos por medio de una computadora. Esto implica lograr la intercomunicación entre aparatos que tienen una lógica de funcionamiento diferente mediante el desarrollo de

los dispositivos que permiten esta intercomunicación. El segundo aspecto se relaciona con el manejo de la información por parte de la computadora para controlar los procesos experimentales en tiempo real, y almacenar los datos relevantes para su posterior análisis. Esto se reduce a un conjunto de instrucciones comprensibles para la computadora para que esta maneje los procesos de entrada y salida de información, efectúe decisiones y almacene en la memoria los datos relevantes. Esto último se logra mediante el desarrollo de un programa de instrucciones en un lenguaje apropiado a las características de la computadora.

El primer aspecto se revisa en el presente capítulo y en el capítulo 2. En el capítulo 3 y 4 se ven los aspectos relacionados con la programación de la computadora para lograr el control de los procedimientos experimentales, almacenamiento y análisis de los datos generados por el experimento.

En este capítulo se analizará brevemente como funcionan los aparatos eléctricos y electromecánicos comúnmente usados en el Análisis Experimental de la Conducta, una descripción general de la arquitectura de una microcomputadora y las características que deben cumplir los dispositivos (interface) que permitan la intercomunicación entre los aparatos eléctricos y los circuitos electrónicos de acceso al usuario en una microcomputadora.

Esta revisión no pretende ser exhaustiva sino tan solo ilustrativa para futuros análisis que se requieran para adecuar la conexión de otros aparatos a la interface. Por tanto, tan solo se describe el funcionamiento eléctrico de algunos dispositivos comunes de una caja de Skinner.

A) Focos: La activación de un foco se logra mediante la aplicación de un voltaje (28 volts) el cual hace que circule una intensidad de corriente eléctrica por una resistencia óhmica, la que disipa una potencia que en el caso del foco se transforma en calor y este al estar en un ambiente de vacío se transforma en energía luminosa.

El diagrama básico se muestra en la Figura 1.1.

Fig. 1.1 : Diagrama de un foco.

B) Relevadores: Son aparatos electromecánicos basados en el principio de los electroimanes. Este principio da cuenta de la atracción o rechazo de cualquier elemento metálico ubicado en la dirección perpendicular de un núcleo ferromagnético envuelto por una bobina al momento en que se aplica una corriente eléctrica a la bobina. Mediante la adecuada disposición de un brazo mecánico, que al ser atraído o rechazado por el núcleo ferromagnético, se puede operar un aparato mecánico conectado a este brazo, tales como los dispensadores de comida para palomas.

Un diagrama de este tipo de aparatos se aprecia en la Figura 1.2.

Fig. 1.2 : Diagrama de un relevador.

C) Interruptores (Microswitch): Están compuestos de tres platinos, dos fijos y uno móvil ubicado entre los dos platinos fijos. Los platinos están unidos a unos brazos que tienen terminales las que permiten su conexión a un circuito eléctrico. Al brazo del platino móvil se puede conectar mecánicamente una tecla, una puerta batiente, una palanca u otro aparato similar.

La condición estable de un interruptor implica el contacto permanente de un platino fijo con el platino móvil. Al estar en contacto habilita la circulación permanente de una corriente eléctrica. Esto significa que el circuito está cerrado y a esta condición se le conoce como "normalmente cerrado". De esto se deduce que existe una condición de circuito abierto entre la terminal del otro platino fijo y la terminal del platino móvil.

En la Figura 1.3 se presenta un diagrama de un interruptor conectado con una tecla en la condición "normalmente cerrado".

Fig 1.3: Interruptor "normalmente cerrado".

En los laboratorios del Análisis Experimental de la

Conducta los interruptores están conectados a diversos elementos ante los cuales los sujetos efectúan sus respuestas, y se utilizan para detectar cambios de voltaje o interrupción del paso de la corriente eléctrica por un circuito dado, como consecuencia de la ejecución de una respuesta.

El accionamiento de los focos y relevadores es básicamente igual y consiste en aplicar un voltaje el cual hace circular una corriente eléctrica por una resistencia o una bobina para accionar un foco en el primer caso o activar un brazo mecánico en el segundo. Usualmente se utilizan estos aparatos u otros de funcionamiento eléctrico semejante para la presentación de estímulos en los experimentos del Análisis Experimental de la Conducta.

La activación de un interruptor mediante la manipulación de una tecla, palanca u otro aparato mecánico implica cerrar un circuito para que el voltaje presente en las terminales del interruptor registre un cambio. La detección de este cambio de voltaje es lo que define a una respuesta.

En suma, la operación de los aparatos descritos anteriormente es lograda mediante la administración o supresión de un flujo de corriente eléctrica o bien detectar el paso, o supresión del paso de corriente eléctrica por un circuito dado.

Si se pretende manejar estos aparatos con una microcomputadora cuyo accionar es electrónico y por ende sus

bases, sus niveles de voltaje y amperaje, y modo de operación difieren de lo eléctrico, es preciso emplear un dispositivo (interface) que adecúe tanto las señales de la computadora a las características propias de estos aparatos eléctricos como las señales generadas por los aparatos electromecánicos a un modo de información compatible con la procesada por la computadora.

A continuación se revisan someramente algunas características del funcionamiento de una computadora. Esto con la finalidad de lograr un mejor entendimiento de las funciones a desarrollar por la interface.

#### ARQUITECTURA BASICA DE UNA MICROCOMPUTADORA

Una microcomputadora esta compuesta por una unidad de procesamiento central (CPU) la que contiene un microprocesador que define las características principales de la microcomputadora. El microprocesador, que es un conjunto de circuitos electrónicos, controla las operaciones lógico-aritméticas, una unidad de control, las unidades de entrada y salida de información y la memoria del sistema. La Figura 1.4 muestra un diagrama de bloques de la mc Commodore 64 (C-64).

Fig. 1.4: Diagrama de bloques de la Commodore 64.

Las operaciones lógico-aritméticas son efectuadas por

una unidad contenida por el microprocesador llamada Unidad Lógica Aritmética y por ser estas operaciones las más conocidas no se entrará en mayores detalles.

La unidad de control es una base de tiempos que genera los ritmos o pulsos de reloj de los diferentes elementos del sistema y cuya utilización en un sinnúmero de procesos excede el ámbito de este trabajo.

Las unidades de entrada y salida permiten al microprocesador comunicarse con el exterior. Por medio de estos dispositivos es posible enviar datos a la computadora ya sea mediante el teclado, unidades de disco u otros aparatos. Además es posible que la computadora envíe datos al exterior mostrando el producto de un proceso ya sea a través de una terminal de video, una impresora u otros.

La memoria del sistema está compuesta por un sistema de circuitos electrónicos que conforman unidades de almacenamiento de datos. En las unidades de estos circuitos (llamadas bytes) se pueden registrar o leer datos utilizados o generados por los procesos de la microcomputadora.

Toda la información requerida o generada por la microcomputadora se almacena en bytes. Un byte es la cantidad de espacio requerido para almacenar un caracter alfanumérico tal como la letra "A", el número "5" o el símbolo "+", o el espacio requerido para guardar un valor igual o inferior a 255 (esto es válido para microcomputadoras que operen con bytes de 8 bits -  $255=2^8$ ). El tamaño de la memoria de una computadora se expresa en

términos del número de bytes que pueda abarcar. Un K o kilobyte es igual a 1024 bytes o  $2^{10}$  bytes.

Para poder registrar un carácter en un byte, el carácter debe de ser codificado en términos de la notación binaria (sistema de notación con base de 2 en vez de 10 como la que usa el sistema decimal). El sistema binario solo tiene 2 dígitos: el cero y el uno. La figura 1.5 muestra los equivalentes binarios para los primeros 10 decimales.

decimal	binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Fig. 1.5: Equivalencia de la notación decimal y binaria.

Dado que el sistema binario tiene solo dos dígitos, estos pueden ser representados electrónicamente. Por ejemplo, un voltaje alto en un circuito se puede utilizar para representar un "1" y la ausencia de voltaje representa un "0". Este sistema de información que maneja solo dos estados es el único que puede procesar la microcomputadora. Toda información diferente a esta modalidad debe de ser transformada por medio de convertidores adecuados,

periféricos al microprocesador tales como convertidores analógicos-digitales, convertidores digitales-analógicos, interfaces del monitor, etc.

Un byte puede estar compuesto por 4, 8, 16, 32 o más bits. La norma que se ha establecido al respecto es que un byte esté compuesto de 8 bits o "palabra" de 8 bits. Un bit, que es la unidad básica que maneja una mc, corresponde a un dígito binario y sólo puede representar un "0" o un "1". Los bits dentro de un byte están ordenados manteniendo siempre la misma posición, de los bits menos significativos (de valores más bajos) a los bits más significativos (de valores más altos). Cada byte en la memoria de una computadora esta identificado por un número y ésa es su "dirección" o posición. En la figura 1.6 se representa esquemáticamente el ordenamiento de los bytes y sus bits. A la izquierda de la figura aparece la dirección de cada byte y en el extremo derecho el valor decimal del byte de acuerdo a lo que está registrado en cada bit del byte.

dirección del byte	+ signif.					-signif.			valor decimal del byte
	B I T S								
	7	6	5	4	3	2	1	0	
0	1	0	1	1	1	0	0	1	185
1	0	0	0	0	1	0	0	0	8
2	0	0	0	0	0	1	1	1	7
3	0	0	1	1	0	0	0	0	48
4	1	0	0	0	0	0	0	0	128
5	0	0	1	1	1	1	1	1	63

Fig. 1.6: Estructura de un byte.

Los datos representados por un byte son de dos tipos: numéricos y alfanuméricos (letras, símbolos...). Los datos numéricos se pueden transformar directamente a binarios y guardarse en la memoria. Para transformar un dato alfanumérico a binario es preciso utilizar un código. Dado que un carácter alfanumérico requiere de más de un bit, el código a utilizar dependerá de cuantos bits puede manejar el microprocesador. El código empleado comúnmente por la microcomputadora es el ASCII (American Standard Code for Information Interchange). Este código emplea 7 bits por carácter y por tanto permite manejar hasta 127 ( $2^7$ ) caracteres distintos y cada uno de ellos puede ser almacenados en un byte.

Cada byte de la memoria del microcomputador tiene una posición determinada y el microprocesador las identifica por su dirección. Por ejemplo el primer byte tiene la dirección 0 decimal y si la microcomputadora tiene 64 K de memoria, la dirección del último byte será 65536 decimal. Cada posición de memoria o bloques de memoria es utilizada para algún propósito particular según sus interconexiones con el microprocesador. El tamaño de la memoria de un computador dependerá del número de bits disponibles en el canal de direccionamiento del microprocesador. En la figura 1.7 se presentan los valores decimales correspondientes a cada bit y en donde se puede notar que el valor decimal es exactamente igual a 2 elevado al número correspondiente de la posición del bit que contiene un "1".

BIT	7	6	5	4	3	2	1	0
VALOR DEC.	128	64	32	16	8	4	2	1

Fig. 1.7: Valor decimal de cada bit.

En la figura 1.8 se presentan algunos ejemplos esquematizados de posibles valores decimales que pueden representar algunos bytes.

Fig. 1.8: Ejemplos de valores decimales representados por diferentes bytes.

De lo anterior se desprende que los datos que maneja una computadora son de dos tipos: "0" y "1", los que en el caso de la microcomputadora C-64 equivalen a los estados eléctricos de 0 volt y +5 volts respectivamente (según la norma TTL).

Esta intensidad de corriente es insuficiente para operar directamente los focos, relevadores y aparatos electromecánicos similares.

Por consiguiente, se requiere de un dispositivo que aumente la capacidad (potencia) de la señal de salida de la C-64 para la presentación de estímulos en un espacio experimental. Para ello es necesario amplificar en intensidad y/o voltaje los 5 volts del "1" lógico a 28 volts con 300 miliamperes para la operación de los relevadores y

para los focos el voltaje requerido permanece sin variación (5 volts) o bien se puede trabajar con focos de 28 volts y la intensidad de la corriente debe aumentarse a 100 miliamperes. La amplificación de intensidad y voltaje se logra con transistores del tipo BJT, los que actúan tanto como interruptores y como amplificadores de voltaje e intensidad a la vez, permitiendo así la activación de los relevadores, focos y aparatos similares.

El detalle de los componentes de los circuitos de salida de la interface se presenta en el capítulo 3.

Para la detección de la operación de un interruptor se utilizó el mismo sistema de comunicación empleado por la C-64 para la lectura del teclado y su concepción se puede ver en el capítulo 3.

Teniendo claro el tipo de transformación de señales que debe efectuar la interface para permitir la comunicación de un microprocesador con los aparatos localizados en uno o más espacios experimentales, es preciso ahora revisar brevemente algunas características de la microcomputadora a emplear para luego, basándose en esas características especificar los circuitos que componen la interface.

## C A P I T U L O   D O S

### DESCRIPCION DE LA INTERFACE:

La interface desarrollada se diseñó para ser usada con la microcomputadora Commodore 64. Además de las razones económicas y técnicas descritas anteriormente para seleccionar la microcomputadora C-64 se tomó en cuenta un criterio adicional consistente en la facilidad de acceso físico a los puertos de entrada/salida de esta microcomputadora. Dado que todo intercambio de información de la M.C. con el medio exterior debe efectuarse por algún canal de comunicación (puerto), se consideró de suma conveniencia que este canal estuviera ubicado en un lugar de fácil acceso físico, sin necesidad de abrir el gabinete de la microcomputadora para su manipulación.

Los puertos de entrada/salida utilizados para el diseño de la interface son las dos terminales de los controladores de juegos (joystick) y el Puerto del Usuario (User Port). Estas terminales son de fácil conexión a un aparato exterior y su programación se ve facilitada dada la amplia información existente sobre sus características técnicas. Además, se eligieron estos puertos para que las entradas a la microcomputadora quedaran separadas físicamente de las salidas de la microcomputadora, facilitando así sus conexiones a equipos periféricos.

Para un mayor entendimiento del equipo desarrollado se estima oportuno describir algunas particularidades de la arquitectura de la C-64 que permitan luego una mayor comprensión de su interacción con la interface y los equipos experimentales a controlar.

## ARQUITECTURA DE LA COMMODORE 64

La mayor parte de la información relacionada con la Commodore C-64 se ha obtenido de los textos "The anatomy of the Commodore 64", "Commodore 64 programmer's reference guide", "Mapping the Commodore 64" y "Commodores 64, guía del usuario".

Las computadoras básicamente tienen tres habilidades: pueden efectuar cálculos, pueden tomar decisiones y pueden comunicar. Las operaciones de cálculo son tal vez las más fáciles de entender y programar dado que las reglas de las matemáticas nos son relativamente familiares. La toma de decisiones tampoco es difícil puesto que las reglas de la lógica son relativamente pocas. Las reglas de la comunicación son, quizás, las más complejas en la medida que implican un amplio conjunto de reglas y a veces poco exacto. Estas reglas permiten la suficiente flexibilidad para comunicarse virtualmente con casi cualquier aparato externo y en una variedad grande de modos. La única norma que es necesario tener presente es la que señala que toda información que se envíe tiene que poder ser recibida y

entendida por quien la recibe.

Por lo dicho anteriormente, aquí interesa aportar algunos elementos para poder entender los procesos de comunicación de la C-64 con aparatos periféricos. Básicamente solo se describen aquellas cuestiones que son relevantes para el presente trabajo.

La figura 1.4 del capítulo 1 es un diagrama de bloques del diseño de la arquitectura de la C-64. En ella se muestran las relaciones entre los diversos elementos del sistema. Se puede apreciar en ese diagrama que la comunicación del microprocesador con el exterior siempre está mediada por algún tipo de dispositivo. De todos estos elementos interesa describir algunas características de los dos controladores de puertos de entrada/salida denominados CIA 6526 (Complex Interfaz Adapter) quienes son los que permiten la comunicación del microprocesador 6510 con aparatos periféricos, tanto para enviar como para recibir información. Cada CIA está configurado por dos canales (puertos) de entrada/salida de 8 bits (líneas) cada uno. Los canales reciben las denominaciones de Data Port A (DPA) y Data Port B (DPB). Cada uno de los 8 bits de cada Data Port tiene la particularidad de ser programado en forma independiente para que transmita información de entrada o de salida del sistema.

Fig. 2.1: Diagrama de los Data Port.

En la figura 2.1 se presenta un esquema del CIA 6526 mostrando los 8 bits del Data Port A y del Data Port B.

Cada CIA es accedido para su control a través de diferentes posiciones de memoria. Las posiciones de memoria que controlan el CIA #1 están localizadas entre los bytes 56320 al 56335 (decimales) y las del CIA #2 están localizadas entre los bytes 56576 al 56591 (decimales).

Fig. 2.2: Diagrama parcial de circuitos de la C-64.

En la figura 2.2 se muestra un diagrama parcial de las conexiones de los CIA 6526 números 1 y 2 a los diferentes elementos periféricos (teclado, controladores de juegos, conector de la casetera, puerto serial, puerto del usuario, etc.,).

A través del CIA # 1 se logra tanto la lectura de datos como el envío de datos a los controladores de juegos (joysticks), a las paletas de control de juegos (paddles) y la lectura permanente del teclado. Además esta pastilla tiene dos relojes y otras líneas de control que se utilizan en múltiples tareas auxiliares para el manejo de datos y procesos.

El CIA # 2, que en su arquitectura es idéntico al # 1, permite el control del puerto serial, la interface serial RS-232 y el puerto del usuario.

Del CIA # 1 y del CIA # 2 interesa describir su esquema

general, los DPA Y DPB y las posiciones de memorias en donde están localizados los DDR (Data Direction Register). La función de los DDR es la de establecer la dirección de los datos que fluyen a través de los puertos DPA y DPB. Mediante la programación de los DDR se determina si los bits de los puertos fungirán como líneas de salida o entrada de información al sistema.

#### DESCRIPCION DEL C.I.A. 6526 NUMERO 1.

Los ocho bits de cada data port del CIA # 1 se usan para construir una matriz de 8 bits (8 x 8) a la cual se conecta el teclado, tal como se aprecia en la figura 2.2. La equivalencia en el teclado de cada celda de la matriz se presenta en la figura 2.3.

Fig. 2.3: Matriz de 8 x 8 bit.

Para que el microprocesador 6510 pueda leer el teclado, la condición inicial (default) del sistema establece mediante los DDR que todos los bits del DPA fungirán como salidas y los 8 bits del DPB como entradas, de tal modo que todos los bits del DPA están en "1" (Byte 56322,255) y los del DPB en "0" (Byte 56323,0). Al establecer esta condición, la lectura de un "0" en cualquier bit significará que la tecla correspondiente a la intersección de esa columna y renglón esta siendo presionada, mientras que un "1" indica

que no está siendo presionada (ver la lógica de los operadores booleanos en el capítulo 3). Un valor de 255 decimal en el byte de lectura del teclado indica que ninguna tecla esta activada. El Sistema Operativo dispone una lectura de estas posiciones de memoria cada 1/60 de segundo.

Estos mismos data ports se utilizan para la lectura de los controladores de juego conectados al Control Port 1 y Control Port 2. El Control Port 1 está conectado en paralelo al DPB del CIA # 1 y el Control Port 2 está conectado en paralelo al DPA del CIA # 1, tal como se puede apreciar en la Figura 2.4.

Fig. 2.4: Conexiones del CIA 6526 #1.

Cada Control Port utiliza los 5 bits menos significativos o de valores más bajos de los 8 bits de cada Data Port. La lectura de las posiciones de memoria que controlan los Control Port mostrarán un "0" en aquellos bits que estan siendo activados y un "1" para indicar que el interruptor del Control Port no está activado.

Dado que las mismas líneas de los Data Ports del CIA #1 se utilizan tanto para el control del teclado como para el manejo de los Control Port, es posible que exista algo de confusión. La rutina del Sistema Operativo que lee el teclado no tiene como discriminar si un bit llevado a "0" fue ocasionado por la presión de una tecla o la operación de

un interruptor del controlador de juegos. Por ejemplo, presionar la tecla del número 2 y mover el controlador de juegos del Control Port 1 hacia la derecha producen el mismo resultado, es decir, ponen en "0" el tercer bit del DPB. Posteriormente se verá como es posible, mediante programación, deshabilitar la rutina de lectura del teclado para que solamente se lean los datos provenientes de los Control Ports y no haya interacción posible con la operación del teclado.

En la Tabla 1 se presentan las funciones de cada bit del DPA y del DPB y sus correspondientes posiciones de memoria.

Tabla 1: Funciones de los bits del DPA y DPB del CIA #1.

Finalmente del CIA # 1 interesa describir brevemente las funciones de los 2 registros que direccionan los datos (Data Direction Register, DDR), de los bits de los Data Ports. Estos dos DDR controlan la dirección de los datos que fluyen sobre el DPA y el DPB. Cada bit del DDR tiene una correspondencia con un bit del Data Port y determina la dirección, ya sea de entrada o de salida del sistema, del dato transmitido por esa línea del Data Port. Si el bit del DDR se lleva a "1", el bit correspondiente del Data Port fungirá como dato de salida del sistema. Por ejemplo, el bit 7 del DDR A controla el bit 7 del DPA y si éste se pone en

"0", el bit 7 del DPA será usado para registrar datos de entrada al microprocesador 6510; y si el bit del DDR es puesto en "1", entonces el bit 7 del DPA se usará para enviar datos al exterior.

La condición inicial (default) del sistema establece que todos los datos del DDR A están en "1" y por tanto todos los bits del DPA funcionan como salidas, y todos los bits del DDR B están en "0" y por ende las líneas del DPB son de entrada. Esto corresponde con lo descrito al hablar de la matriz utilizada para la lectura del teclado.

La tabla 2 muestra la posición de memoria para el DDR del Data Port A y el DDR del Data Port B del CIA # 1 y su correspondencia con los bits de los Data Port.

Tabla 2: Funciones de los bits del DDR-A y DDR-B del CIA #1.

Las demás posiciones de memoria destinadas al CIA # 1 controlan relojes, el puerto serial y líneas de control cuyas propiedades no participan en el diseño y control de los circuitos de la interface y por tanto se prescinde de sus descripciones.

DESCRIPCION DEL C.I.A. NUMERO 2.

El CIA # 2 , que en su arquitectura es idéntico al CIA # 1, está localizado en los bytes 56576 al 56591 (decimales),

y mediante él se controla el puerto serial que se conecta a aparatos tales como impresoras o unidades de disco, la interface serial RS-232 utilizada para telecomunicaciones y el Puerto del Usuario (User Port) que es un puerto paralelo de 8 bits que puede ser utilizado para una diversidad de propósitos. Este último es el que interesa describir puesto que dada su versatilidad, facilidad de programación y acceso es el que se utilizó para el diseño de los circuitos de salida de la interface.

En el caso del CIA # 2 el DPA es utilizado para comunicación con el puerto serial, control de la memoria destinada a la graficación en video, y algunas de sus líneas también participan en la comunicación de la RS-232. Los bits del DPB participan del control de la interface serial RS-232 al conectarse algún cartucho de este tipo al puerto del usuario; en caso contrario los 8 bits del DPB están disponible. Todas las líneas del DPB están accesibles físicamente en el conector exterior del Puerto del Usuario ubicado en la parte posterior del gabinete de la C-64. Estas líneas de transmisión de datos pueden utilizarse para interactuar con aparatos externos ya sea enviando o recibiendo información según lo determine la programación de los DDRs del CIA # 2.

La figura 2.5 muestra un diagrama del puerto del usuario y describe la correspondencia de las líneas de los Data Ports del CIA # 2 con los conectores exteriores del Puerto del Usuario.

Fig. 2.5: Conexiones del DPB del CIA #2.

En la tabla 3 se describen las funciones de cada bit del DPA y del DPB y sus correspondientes posiciones en la memoria del C-64.

Tabla 3: Funciones de bits del DPA y DPB del CIA #2.

Los DDR del CIA # 2 controlan la dirección del flujo de la información a través de las líneas de los DPA y DPB del CIA # 2 de igual manera que en el caso del CIA # 1. La condición inicial (default) para el DDR A es 63 (decimal) en la que todos los bits, a excepción de los bits 6 y 7, son de salida, y para el DDR B es 0 decimal y por tanto todos los bits del DPB fungen como entrada al sistema.

En la tabla 4 se describen las funciones y correspondencia de cada bit de los DDR del Data Port A y DDR del Data Port B del CIA # 2 y sus localizaciones en la memoria de la C-64.

Tabla 4: Funciones de bits del DDR-A y DDR-B del CIA #2.

REQUERIMIENTOS DE LA INTERFACE

La microcomputadora debe permitir la programación de eventos y el registro de respuestas ocurridas en un espacio en el mismo momento en que ellas ocurran. Esto implica, hablando en términos propios del Análisis Experimental de la Conducta, por una parte, disponer la presentación de estímulos bajo ciertas condiciones estipuladas de antemano y, por otro lado, registrar la ocurrencia de cualquier respuesta que ocurra en un espacio predeterminado. Se entiende por presentación de estímulos a la activación de un aparato eléctrico (focos, parrillas de choque eléctrico, etc.) o electromecánicos (dispensadores de reforzadores, proyectores de diapositivas, etc.) y por respuesta se entiende el cierre de un microswitch (teclas, palancas, etc.) o el cambio de estado de una condición dada (células fotoeléctricas).

La presentación de estímulos, en la medida que se deben operar aparatos externos a la microcomputadora, debe ser controlada por líneas de salida de la microcomputadora y mediada por una interface que permita la correcta comunicación de los datos necesarios para el control de los aparatos externos por parte de la microcomputadora. Y las respuestas ocurridas en el medio externo a la microcomputadora deben transformarse, mediante la interacción con la interface, en datos de entrada comprensibles para la microcomputadora. De ello se desprende que la función primordial de la interface es la adecuada transformación de los datos provenientes de la

microcomputadora y de los datos enviados a la microcomputadora para que ésta pueda interactuar correctamente con aparatos externos.

En el Análisis Experimental de la Conducta es común manejar una gran variedad de estímulos y simultáneamente se registran una cantidad menor de respuestas. Esto, de alguna manera determina que la interface debe contar con una mayor cantidad de salidas (para la presentación de estímulos) que de entradas (para el registro de respuestas).

Teniendo presente estas condiciones se determinó utilizar los controladores de juego 1 y 2 (Control Port 1 y Control Port 2) para la recepción de hasta 5 respuestas (independientes o simultáneas) en cada uno de ellos. Por lo tanto, mediante la utilización de la interface y las adecuadas conexiones y programación, es posible registrar la ocurrencia en tiempo real de hasta 10 respuestas (simultáneas o no) ocurridas en un espacio experimental. Además, mediante programación y la adecuada conexión, es posible distribuir estas 10 respuestas en espacios experimentales diferentes permitiendo así la operación de más de una caja experimental a la vez. El registro de las respuestas en tiempo real se puede obtener al incluir la lectura de uno de los relojes de la C-64 al momento de leer las posiciones de memoria de los control ports, registrándose de esta manera tanto la ocurrencia de alguna respuesta como el tiempo en que ella acaeció.

Para la presentación de estímulos se optó por utilizar

el Puerto del Usuario para que mediante el uso de las líneas del DPB del CIA # 2 se contaran con 16 líneas de salida para el control de aparatos externos ubicados en uno o más espacios experimentales. La elección del Puerto del Usuario se realizó considerando la facilidad de programación de cada uno de los bits del DPB ya sea como entrada o salida (en este caso todos se programaron para funcionar como salidas). Además, se utilizan dos líneas del DPA para ampliar la capacidad del puerto del Usuario al multiplexar los 8 bits del DPB del CIA # 2. La multiplexación permite a varios elementos compartir las mismas líneas de comunicación en diferentes momentos. En este caso particular, los 8 bits del DPB se utilizan para controlar 8 líneas de salida en un momento y otras 8 líneas de salida en otro momento, contando de esta manera con hasta 16 líneas de salida para controlar igual número de estímulos.

De esta manera se estableció, entonces, el diseño de una interface que permitiese la utilización de 16 líneas de salida de la microcomputadora para el control de hasta 16 estímulos ubicados en uno o más espacios experimentales, teniendo en cuenta que ellos puedan ser activados con diferentes voltajes o amperajes, y el registro de hasta 10 respuestas (simultáneas o no) que pueden ocurrir en espacios experimentales distintos.

CONFIGURACION DE LA INTERFACE

De acuerdo con los requerimientos planteados anteriormente, la interface debe contar con 16 líneas de salida para la programación de estímulos y con 10 líneas de entrada para la detección y registro de respuestas. A continuación se describen los circuitos electrónicos construidos para lograr tal objetivo con la microcomputadora Commodore 64.

Las salidas de la interface:

Los circuitos de salida de la interface están compuestos por tres tipos de elementos: un multiplexador, cuatro memorias de 4 bits cada una y 16 transistores, tal como se puede apreciar en la figura 2.6. En seguida se describen las características y funciones de cada uno de estos elementos.

Fig. 2.6: Circuitos de salida de la Interface.

a) El multiplexador: Al multiplexar una línea (bit) se comparte su uso para manejar la misma o diferente información, la que interactúa con distintas líneas periféricas. La multiplexación permite a varios elementos compartir las mismas líneas de comunicación usándolas en diferentes momentos. Esta función es coordinada por un circuito integrado especial al que simplemente llamaremos multiplexador. Su número de identificación es el chip # 7408. Este multiplexador permite que un bit control al estar

en "1" haga fluir la información de los bits que controla hacia las líneas con que esté conectado y cuando está en "0" impide el flujo de información.

Para conformar las 16 líneas de salida se determinó utilizar los 8 bits del Data Port B del CIA 6526 #2. Para ello fue necesario ampliar estas 8 líneas de salida de tal modo que los mismos 8 bits fuesen usados en forma alternada: esto es, en una ocasión se utilizan las líneas del DPB para enviar 8 bits de información al puerto 1 de salida de la interface y en otra instancia se emplean las mismas 8 líneas del DPB para enviar otros 8 bits de información al puerto 2 de salida de la interface. Para controlar la alternancia en el uso de los 8 bits del DPB se usan los bits 2 y 3 del Data Port A del mismo CIA 6526 # 2. El bit 2 corresponde al conector "M" del puerto del usuario y el bit 3 al conector "9".

El multiplexador (chip # 7408) está compuesto por cuatro compuertas AND. Dos de estas compuertas se conectaron por un lado a la línea DPA 2 y por otro lado a dos chips de memoria. Las otras dos compuertas están conectadas a la línea DPA 3 y a otros dos chips de memoria. Al usar dos compuertas se estableció una lógica combinatoria que permite la programación independiente o simultánea de las líneas de salida de la interface.

En la Figura 2.7 se muestra una tabla de correspondencia de los valores de los bits 2 y 3 del DPA del CIA 6526 # 2 con la ejecución de los puertos 1 y 2 de salida

de la interface. Es preciso hacer notar que en el caso del bit 3 del Data Port A un "0" fungirá como "1" puesto que esta línea tiene un inversor tal como se aprecia en la figura 2.2 del diagrama de la microcomputadora. Y un valor de "1" tendrá un valor funcional de "0".

BIT	VALOR	EJECUCION DE LOS PUERTOS
2	"0"	NO ACTUA EL PUERTO 1
3	"0"	ACTUALIZA LA INFORMACION DEL P2
2="0" y 3="1"		NO ACTUA NINGUN PUERTO
2	"1"	ACTUALIZA LA INFORMACION DEL P1
3	"1"	NO ACTUA EL PUERTO 2
2="1" y 3="0"		AMBOS PUERTOS ACTUAN IGUAL

Figura 2.7: Correspondencia de los bits control del multiplexador.

Lo básico de esta lógica es que al existir un "0" en el bit control no se afecta la información grabada en las memorias de las líneas de salida de la interface. Al estar en "1" el bit control, se actualiza la información de las memorias. Esto está ya indicando que la información transmitida por el multiplexador es enviada a unas memorias que permiten retener las señales que van a controlar los aparatos en los espacios experimentales.

b) Las memorias: Fue preciso emplear unas memorias para poder mantener constante la acción de la información enviada a cada línea de salida de la interface. Se utilizaron cuatro memorias de 4 bits cada una, identificadas como chips #

7475. Estas son memorias de lectura y escritura; usualmente las memorias deben ser habilitadas tanto para poder grabar datos como para poder leer esos datos. En este caso, la misma señal se utiliza para ambos fines puesto que las terminales de escritura (W) y de lectura (R) están conectadas a la misma línea, tal como se puede apreciar en el diagrama de la interface. De este modo, los datos escritos en la memoria son de inmediato traspasados a las líneas de salida de la interface y mantenidos constantes para que su acción en los aparatos periféricos sea estable hasta que así lo disponga la condición programada.

Se usaron cuatro memorias de cuatro bits cada una para conformar una salida de 16 bits. Las memorias están dispuestas en bloques de dos memorias para configurar una salida de 8 bits para cada puerto de la interface. Cada memoria es actualizada con la información disponible en el DPB y coordinada por el multiplexador. Esto es posible puesto que las entradas de las memorias están conectadas en paralelo a las salidas del DPB.

Las salidas de las memorias están conectadas a la base de un transistor cuya función es doble: actúa tanto como un interruptor como un amplificador de potencia de la señal.

c) Los transistores: En la figura 2.8 se presenta un diagrama básico de un transistor, señalándose sus tres partes: base, colector y emisor. En este caso se utilizaron transistores tipo NPN Darlington TIP 110.

Fig. 2.8: Diagrama de un transistor.

A la base de cada transistor llega una línea de salida de la memoria. Si la línea de la memoria tiene un "1", ese "1" pasa a la base del transistor ocasionando que el circuito se cierre (actuando como un interruptor) y el colector queda con un voltaje de 0 volt y se produce una circulación de corriente activándose el aparato que esté conectado al producirse una amplificación (función amplificadora) de la potencia de la señal transmitida. Al existir un "0" en la salida de la memoria el circuito permanece abierto y por lo tanto el colector y el dispositivo tienen igual voltaje lo que implica que no hay conducción de corriente y por ende no se activa el aparato conectado al colector del transistor.

El colector del transistor corresponde a la terminal de salida de la interface y es a esta terminal en donde se debe efectuar la conexión de un aparato tal como un foco, dispensador de alimentos, dispensador de choques, etc. Gracias a la acción del transistor se pueden activar equipos que funcionen hasta con 28 volts y 1.5 amperes.

Fig. 2.9: Conectores de la Interface.

La figura 2.9 representa el aspecto exterior de la interface mostrando sus conectores de salida de cada puerto,

los conectores de los dos puertos de entrada y los conectores de alimentación de la interface y la conexión a tierra.

Las entradas a la interface:

Las entradas de la interface simplemente son unas terminales que facilitan la conexión a aparatos periféricos, puesto que estas líneas están conectadas directamente a los control port 1 y control port 2 de la microcomputadora. De cada control port se utilizan los 5 bits más bajos, y la correspondencia de cada bit es idéntica a la señalada en los puertos de entrada de la interface. Esto significa que el bit 0 del control port 1 corresponde al bit 0 del puerto de entrada número 1 de la interface.

#### CONEXIONES DE LA INTERFACE

Son cuatro tipos de conexiones las que hay que considerar para la interface.

a) Alimentación de la interface:

Se requiere de una fuente de poder que suministre +5 volts a 0.5 amperes para el funcionamiento de la interface. En una versión posterior se incorporó una fuente interna a la interface la que permite transformar la energía de 28 volts comúnmente usada para la mayoría de los aparatos a la energía requerida por los circuitos de la interface.

La conexión de los +5 volts debe efectuarse al conector señalizado con "+5 v" en el Puerto 1 de salida de la interface. La conexión a tierra debe realizarse al conector claramente señalado en las terminales del Puerto 1 de la interface.

b) Conexión de las salidas:

La mayoría de los aparatos a controlar con la interface requieren de un mayor voltaje y/o amperaje que el proporcionado por las líneas de la microcomputadora. Por ejemplo, un dispensador de alimentos puede necesitar 28 volts y hasta dos amperes. Por lo general, es preciso usar una fuente externa para la operación de los aparatos comúnmente usados para la presentación de estímulos en una caja de Skinner.

Fig. 2.10: Conexión de un foco.

En la figura 2.10 se exhibe un diagrama de la conexión de un foco a una terminal de salida de la interface. Nótese que el foco debe ser alimentado con una fuente externa de +28 volts, de las que usualmente se encuentran en los laboratorios de psicología. El modo de operación de la interface establece que si el bit correspondiente a la terminal de salida está en "0", el circuito del transistor permanece abierto y no hay conducción de corriente y por tanto el foco permanece apagado. Si, en cambio, ese bit está

en "1" se cierra el circuito, permitiendo el flujo de corriente y encendiendo el foco.

La lógica de la conexión de cualquier aparato que opere como estímulo es similar: en un polo se requiere de una fuente externa para su alimentación y el otro polo se conecta a una terminal de salida de la interface, la que fungirá como interruptor para la activación del aparato. El aparato permanecerá activado mientras así lo determine la condición programada. Para el caso de aparatos que se operen con un pulso eléctrico (dispensadores de comida, dispensadores de choque eléctrico) la única diferencia es que por programa se debe controlar la duración de la señal transmitida para que su acción sea discreta y no continua.

#### c) Conexión a las entradas:

Puesto que para las entradas se están utilizando los dos controladores de juegos (control port 1 y control port 2) de la microcomputadora, es necesario tener presente que los aparatos que se utilicen para el registro de respuesta por lo comun no requerirán de fuente de poder externa dado que la misma microcomputadora proporciona la energía requerida. Como se veía anteriormente, los controladores de juego estan conectados en paralelo al teclado y éste a su vez a una matriz de 8 por 8 bits compuestos por los DPA yDPB del CIA 6526 # 1. Los 8 bits del DPA están siempre en "1", es decir, envían permanentemente una señal de +5 volts y los 8 bits del DPB siempre están en "0". Bajo esta condición

los controladores de juegos siempre están enviando una señal de "1" al microprocesador. Al cerrarse un circuito conectado a una terminal de entrada de la interface la señal enviada a su data port sera un "0".

Fig. 2.11: Conexión de un interruptor.

Comúnmente lo que se utiliza para detectar respuestas son relevadores y para un adecuado funcionamiento el polo normalmente abierto debe conectarse a la terminal de entrada de la interface y el común debe ir a tierra. En la figura 2.11 se muestra un ejemplo de conexión de un relevador al contactor de entrada de la interface. En el caso de los detectores de lamidas a un bebedero, es aconsejable utilizar un circuito amplificador inversor alimentado con fuente externa de +5 volts dada la alta resistencia presentada por la rata. Un diagrama de este tipo de circuito se presenta en la figura 2.12.

Fig. 2.12: Circuito amplificador inversor.

Es recomendable que este circuito se encuentre lo más cercano posible al detector de lamidas con el objeto de evitar ruido o interferencias que alteren la señal.

d) Conexión de la interface a la microcomputadora:

La interface cuenta con un cable que contiene las 16

líneas de salida los que terminan en un conector "hembra" que corresponde a la terminal "macho" del puerto del usuario de la C-64. Se debe respetar la marca que señala el lado que debe ir hacia arriba en este conector. Las líneas de entrada de la interface están incorporadas a dos conectores "hembras" que complementan las terminales de cada control port de la C-64. Los conectores de la interface están debidamente identificados para su adecuada conexión al control port 1 y al control port 2.

#### PRUEBA DE LA INTERFACE

Una vez efectuadas las conexiones de la interface a la fuente de + 5 volts, a la tierra y a la C-64 se puede proceder a probar el correcto funcionamiento de las salidas y entradas de la interface.

a) Prueba de las salidas de la interface: Con un voltímetro se puede verificar que todas las salidas de la interface deben estar entregando 0 volts al estar encendida la C-64, y la fuente de +5 volts.

Para comprobar que la interface esté trabajando en orden se puede utilizar un foco de +5 volts conectado con un polo a la fuente de +5 y con el otro (tierra) se irá probando cada terminal de salida de la interface. Con la siguiente instrucción cada vez que el cable de tierra del

foco haga contacto con una terminal de salida se deberá de encender.

POKE 56578,63:POKE 56579,255:POKE 56576,23:POKE 56577,255

Si la instrucción anterior es reemplazada por la siguiente, en todas las terminales de los dos puertos de salida el foco deberá permanecer apagado.

POKE 56578,63:POKE 56579,255:POKE 56576,23:POKE 56577,0

En caso de no observarse lo dicho anteriormente se deberá a) revisar la fuente de +5 volts, b) revisar las conexiones de la interface a la fuente y a la C-64. En caso de que sea una línea particular la que esté dando problemas se deberá revisar el fusible de esa línea y el transistor de la misma. Ambos elementos son de fácil sustitución.

Prueba del funcionamiento de las entradas: Conectada la interface a la fuente de +5 volts, a la tierra y a la C-64 se procederá a correr el programa presentado en la rutina 2 modificando el renglón 96 por un GOTO 205.

Con un cable conectado a tierra se establece contacto con una de las terminales de entrada a la interface. Al efectuarse el contacto se deberá de presentar en pantalla el número del bit y la identificación del Control Port activado.

Si no se observa respuesta alguna se deberá a) revisar que cada línea de entrada este entregando +5 volts en la condición estable, b) revisar la fuente de +5 volts, c) revisar todas las conexiones.

Si un bit particular presenta problemas se deberá revisar que exista una correcta conducción entre el conector de la interface y la terminal "hembra" que embona con la terminal del Control Port de la C-64.

## C A P I T U L O   T R E S

### PROGRAMACION DE LA INTERFACE.

Para efectos de programación de los eventos a controlar por medio de la interface es preciso realizar un programa de instrucciones en un lenguaje compatible con las características de la microcomputadora C-64. Este programa puede estar escrito en lenguaje BASIC, en ensamblador o lenguaje de máquina. La ventaja de estos dos últimos lenguajes es su alta velocidad de ejecución y su desventaja es la relativa complejidad para escribirlo y eventualmente modificarlo. Para las necesidades típicas de un laboratorio de psicología creemos que el lenguaje BASIC es el más indicado por su facilidad de programación, accesible a cualquier investigador, y con un compilador apropiado se puede aumentar su velocidad de ejecución en forma notable.

Cualquiera que sea el lenguaje de programación que se seleccione, es preciso conocer las especificaciones requeridas por la interface para programar sus salidas y detectar y registrar las señales de entrada.

Los ejemplos que aquí se proporcionan están dados en lenguaje BASIC suponiendo que quien quiera realizar sus programas en un lenguaje de más bajo nivel con esta información podrá escribir sin problemas sus propias rutinas.

### PROGRAMACION DE LAS SALIDAS DE LA INTERFACE:

Cada una de las 16 salidas de la interface puede ser programada en forma independiente para manejar diversos aparatos usados comúnmente para la presentación de estímulos en una caja de Skinner. Al contar con 16 líneas se puede optar por controlar simultáneamente el desarrollo de dos sesiones experimentales, o bien una sola caja experimental compleja.

La programación de cada una de las salidas se efectúa por medio del programa escrito en la microcomputadora C-64. Para ello es necesario: 1) Habilitar el Data Port B del CIA 6526 # 2 para que todos sus bits funcionen como líneas de salida, 2) establecer mediante la manipulación de los bits control DPA 2 Y DPA 3 el direccionamiento de la información a uno de los dos puertos de salida de la interface o a ambos a la vez, y 3) Especificar cuales líneas de cada puerto son las que se van a activar y cuales van a ser inoperativas.

Para cualquier manipulación que se quiera efectuar en las diferentes direcciones de memorias (DDR, Data ports, control ports) es preciso realizarlas con las sentencias PEEK y POKE del lenguaje BASIC. La sentencia PEEK permite leer la información de cada uno de los 8 bits de una determinada dirección de la memoria. En el ejemplo 1, la instrucción PEEK proporcionará por la un número entero comprendido en el rango de 0 a 255, lo que corresponde a 2 elevado a la 8a y que será el valor decimal de los bits

contenidos en la dirección de memoria 56320. (Ver ejemplos de las figuras 1.6 y 1.7).

10 PRINT PEEK (56320)

#### Ejemplo 1

Mediante la sentencia POKE es posible tener acceso a una determinada dirección de la memoria y registrar en ella la información que uno desee. En el ejemplo 2, la línea 10 del programa establece que todos los bits de la dirección de memoria 56323 estarán en "0", mientras que en la línea 20 se dispone que todos los bits de la dirección 56322 estén en "1" y en la línea 30 se ponen en "1" los bits 0,1,2 y 4.

```
10 POKE 56323,0
20 POKE 56322,255
30 POKE 56322,23
```

#### Ejemplo 2

Teniendo claro el modo de manipular los datos contenidos en cada bit de las diferentes direcciones de la memoria, se verá a continuación en detalle la programación de estímulos.

La información requerida para programar cada línea del Data Port A y B del CIA 6526 # 2 está comprendida en el siguiente cuadro:

DATA PORT	DIRECCION	D D R	DIRECCION
DPA	56576	del DPA	56578 "0"=entrada
DPB	56577	del DPB	56579 "1"=salida

Figura 3.1: Dirección de los Data Port A y B y de sus DDRs.

Conforme a lo planteado anteriormente, para programar las líneas de salida de la interface es preciso:

1) Habilidad de las líneas de salida: Esto implica habilitar el Data Port B del CIA 6526 # 2 para que todos sus bits funcionen como líneas de salida. Para ello es necesario manipular los DDR del Data Port B y del Data Port A. Esto se logra con las siguientes líneas escritas en BASIC, del ejemplo 3, las que establecen que los 8 bits de la dirección de memoria 56579 estén en "1", habilitando así como líneas de salida a los 8 bits del Data Port B y la línea 20 habilita a los primeros 6 bits del Data Port A como líneas de salida. De esta forma todo dato enviado al Data Port B siempre serán datos de salida a uno o a ambos puertos de la interface, según lo estipulado en el punto siguiente.

```
10 POKE 56579,255
```

```
20 POKE 56578,63
```

Ejemplo 3.

2) **Direccionamiento** de los Data Port: Esto es establecer el direccionamiento de la información hacia uno de los puertos de salida de la interface o a ambos simultáneamente. Esto se obtiene manipulando los bits control DPA 2 y DPA 3, los que tienen como función la multiplexación de las líneas del Data Port B. Aquí se vuelve a señalar que el DPA 3 tiene un inversor en la línea de salida de la microcomputadora, tal como se aprecia en la figura 2.2. Esto implica que si el bit tiene un "1", el inversor lo transformara en "0" funcional.

Se recordará que si el bit control tiene un "1" permite el paso de la información del Data Port B a las líneas que este bit control coordina. Si el bit control está en "0", impide el flujo de la información. En el cuadro siguiente se presentan las posibles condiciones de la dirección de memoria 56576 que corresponde al Data Port A.

	DATA PORT A DIRECCION 56576								
	B I T S								
	7	6	5	4	3	2	1	0	
Decimal	128	64	32	16	8	4	2	1	
Actua P1	0	0	0	1	1	1	1	1	= 31
Actua P2	0	0	0	1	0	0	1	1	= 19
Act P1/P2	0	0	0	1	0	1	1	1	= 23

Figura 3.2: Direccionamiento de la información a puertos de la interface.

Al activar el Puerto 1 con POKE 56576,31, se está poniendo en "1" el DPA 2 y el DPA 3, pero el DPA 3 al estar con un inversor funge como "0" y por tanto sólo se está

habilitando al DPA 2 como emisor de información y por consiguiente solo el puerto 1 de la interface está siendo actualizado con la información del Data Port B. Con la instrucción POKE 56576,19 la situación es exactamente la opuesta, habilitándose solamente el puerto 2 de la interface, y con POKE 56576,23 se habilitan simultáneamente los dos puertos con la misma información del Data Port B.

3) Activación de líneas: Esto implica el especificar las líneas a operar de cada puerto de salida de la interface. Esto es factible mediante la manipulación de los bits del Data Port B cuya dirección de memoria es 56577. Al poner un "1" en un bit determinado se está disponiendo la activación del aparato que está conectado en la terminal correspondiente de uno o ambos puertos de salida de la interface, según lo que se haya direccionado en Data Port A. Un "0" implica la no activación del aparato conectado a la línea de salida de la interface.

La activación del aparato durará tanto tiempo como dure la condición programada. En el ejemplo 4 se esta habilitando el puerto 1 y los 4 primeros bits están operativos y los 4 bits más altos estan en "0" y por consiguiente los aparatos conectados a esas líneas estarán inactivos.

10 POKE 56576,31:POKE 56577,15

20 POKE 56576,31:POKE 56577,16

Ejemplo 4.

En el renglón siguiente del ejemplo 4 se apagan los cuatro aparatos conectados a las 4 primeras líneas del Puerto 1 y se activa el dispositivo conectado a la 5ª línea solamente.

Es recomendable que al inicio del programa se establezcan aquellas condiciones que van a estar vigentes durante toda la sesión y declarar como variables los valores de aquellas direcciones de uso frecuente así como los valores que asumen los estímulos conectados a cada línea de la interface, asignándoles el valor decimal del bit que le corresponde. Esto se ve aplicado en el ejemplo 5.

#### Rutina 0: Inicialización.

Los variables declaradas en este ejemplo se volverán a emplear en futuras demostraciones. Nótese que en la inicialización de la rutina 0 (líneas 10 a 60), los valores de las direcciones de memoria de uso repetitivo se asignan a variables para simplificar su aplicación posterior. En la línea 70 se asigna a cada variable el valor decimal del bit al cual están conectado los aparatos, de tal modo que para efectos de programación tan sólo sea necesario habilitar el puerto de la interface y explicitar cuales dispositivos son los que estarán operativos en ese momento. En este ejemplo las conexiones están efectuadas de acuerdo a la siguiente tabla y de igual manera para ambos puertos de la interface. Esta tabla de conexiones y las denominaciones de las

variables son las que se utilizan en todas las rutinas de demostración.

BIT	DECIMAL	VARIABLE	ESTIMULO
0	1	LC	Luz Central
1	2	RB	Ruido Blanco
2	4	SR	Disp. de Reforz.
3	8	LR	Luz Comedero
4	16	FZ	Foco Izquierdo
5	32	FC	Foco Central
6	64	FD	Foco Derecho
7	128	CH	Dispensador Choques

#### Ejemplo 5.

En la línea 80 se habilita las líneas del puerto 1 de la interface y se envía la información necesaria para activar LC (luz central), RB (ruido blanco) y FZ (foco izquierdo). En la línea 90 se habilita el puerto 2 de la interface y se activa SR (dispensador de reforzadores), LC (luz del comedero) y RB (ruido blanco). Nótese que si se quieren activar dos o más líneas simultáneamente tan sólo es necesario agregar las variables correspondientes a esas líneas encadenadas con el signo "+".

Para aquellos aparatos que se operan con un pulso eléctrico (dispensadores de reforzadores y de choques eléctricos, proyectora de diapositivas...), es necesario controlar el tiempo de duración de la instrucción que activa dicho aparato, tal como se presenta en el ejemplo de la rutina 5.1. El manejo del tiempo por programa se verá más adelante.

**b) DETECCION Y REGISTRO DE RESPUESTAS:**

Para detectar la ocurrencia de respuestas acaecidas en un espacio exterior se utilizan las entradas de los controladores de juego (control port 1 y control port 2). Cada control port emplea los 5 bits menores de cada Data Port del CIA 6526 # 1, tal como se describe en la figura 2.4 y en la tabla 1 del Capítulo 2. Es necesario tener presente que estas mismas líneas del Data Port A y B se usan para la lectura del teclado de la microcomputadora.

Para poder detectar el cierre de un interruptor o de un relevador es preciso: 1) inhabilitar la lectura del teclado, 2) efectuar una lectura de las direcciones de memoria que regulan a cada control port, 3) identificar las líneas activadas y las no activadas, 4) realizar el registro de las respuestas o no respuestas ocurridas y 5) habilitar el teclado nuevamente, si se requiere de su uso.

A continuación se explicitan los detalles de cada paso enumerado más arriba:

1) Inhibición de la lectura del teclado: En la medida que las líneas de cada control port están conectadas en paralelo a las líneas de cada Data Port del CIA 6526 # 1, mismas que se utilizan para la lectura del teclado, es necesario suprimir momentáneamente la rutina de lectura del teclado para evitar posibles confusiones. Esto se logra anteponiendo la siguiente instrucción POKE 56322,224 a cada lectura de los control port. Para volver a habilitar la rutina de

lectura del teclado habría que agregar la instrucción POKE 56322,255. Si durante toda la sesión no se van a capturar datos por medio del teclado la instrucción de habilitación de la lectura del teclado puede ir hasta una sección final del programa.

2) Lectura de respuestas: Esto implica el efectuar una lectura de las direcciones de memoria de cada control port: Con la sentencia PEEK es posible leer el estado de cada bit de una determinada posición de memoria. La dirección de memoria del Control Port 1 esta localizada en el byte 56321 y la del Control Port 2 en el byte 56320. Entonces para conocer el estado de cada control port es preciso aplicar la sentencia PEEK a cada dirección de memoria.

En el Capítulo 2 se veía que si un aparato conectado a una línea de entrada está activado, el bit correspondiente mostrará un "0" y si está inactivo el bit mostrará un "1". Usualmente se debería destinar un control port para la lectura de hasta 5 operandos de una caja experimental y el otro control port para otros 5 operandos en otra caja, lo que hace prácticamente improbable la ocurrencia de respuestas simultáneas en un mismo control port. En caso de que puedan darse respuestas simultáneas la estrategia de lectura varía ligeramente. La rutina siguiente permite una lectura confiable de cada control port.

Rutina 1: Lectura de Control Port 1.

Para la lectura del control port 2 sería necesario cambiar la dirección de memoria en la línea 205 por 56320.

3) Identificación de las líneas activadas: Realmente lo que se hace es una identificación tanto de las líneas activadas como de las no activadas: La lectura de cada dirección de memoria que regulan los control port proporcionarán una cifra correspondiente al valor decimal de los estados de los 8 bits de cada dirección. Por ejemplo, si ningún operando está activado al momento de leer el puerto, la sentencia PEEK de ese control port entregará la cifra de 255. Para poder identificar el estado de los 5 bits más bajos que configuran cada control port es recomendable utilizar una "máscara", tal como se muestra en la rutina de lectura de respuestas.

En dicha rutina se está utilizando un operador lógico AND para aislar cada bit del control port y facilitar así su lectura. A continuación se presenta la Tabla de verdad de los operadores booleanos, la que permitirá comprender su acción.

La operación AND produce un resultado de "1" sólo si los dos números comparados son "1".

1	AND	1	=	1
0	AND	1	=	0
1	AND	0	=	0
0	AND	0	=	0

La operación OR produce un resultado de "1" si cualquiera de los dos números comparados es "1".

1	OR	1	=	1
0	OR	1	=	1

1 OR 0 = 1  
0 OR 0 = 0

La operacion NOT complementa lógicamente cada número.

NOT 1 = 0  
NOT 0 = 1

Tabla 3.1: Tabla de verdad booleana.

En la rutina de lectura de los control port, en las líneas 220 a 260 se emplea un AND para los primeros 5 bits de tal modo que solo el bit de la línea que esté activada va a mostrar un resultado igual a "0". Por ejemplo, la sentencia 250 determina si está activado o no el bit 3 del control port 1. Supóngase el siguiente caso:

	BIT	7	6	5	4	3	2	1	0
Dirección 56321		0	0	0	1	0	1	1	1
AND 8		0	0	0	0	1	0	0	0
Resultado :		0	0	0	0	0	0	0	0

El resultado de la operación AND muestra un 0, indicando que el operando conectado al bit 3 está activado ( bit 3 = "0") y que al estar "enmascarado" por un AND 8 lo compara con un tercer bit en "1" , cuyo resultado se acuerdo a la tabla de verdad booleana necesariamente es un "0". De esta forma la rutina de lectura de respuestas revisa el estado de cada uno de lo 5 bits del control port y si el resultado del enmascaramiento con la operación AND muestra un "0" ésto se debe interpretar como que el operando

conectado a ese bit está activado.

En el ejemplo siguiente se presenta una rutina que lee los 2 control port y revisa los 5 bits de cada control port:

#### Rutina 2: Lectura de ambos Control Ports.

Existen diferentes formas de leer los puertos y aislar los bits que interesen. La forma más apropiada la determinará los intereses del investigador.

La capacidad de la microcomputadora permite efectuar hasta 13 o 14 lecturas de cada puerto por segundo usando un compilador de BASIC. Sin compilador se pueden lograr hasta 8 lecturas por segundo. Si se requiere de una mayor resolución en la lectura de respuestas será preciso utilizar un ensamblador o lenguaje de máquina para escribir estas rutinas.

La siguiente rutina (rutina 3) muestra un modo de lectura de ambos puertos en donde a cada puerto se han conectado 3 operandos. En esta rutina lo que interesa es detectar que tipo de respuestas, o no respuesta, hubo en cada lectura y se registra el estado de cada lectura. Se efectúan 180 lecturas en cada uno de los 50 ensayos. Un ensayo dura aproximadamente un minuto y se efectúan tres lecturas por segundo. Al término de las 180 lecturas se presenta reforzamiento, independiente de la ejecución del sujeto.

## Rutina 3: Lectura de Puertos.

En este ejemplo se efectúan algunos controles de tiempo. El detalle de estas operaciones se verá más adelante.

4) Registro de las respuestas: La capacidad de almacenamiento de datos de la microcomputadora es finita. Se puede llegar a disponer de alrededor de 40 K de memoria al tener acceso a los 4 K no utilizados por el sistema operativo de la C-64. Por tanto, para optimizar su uso se debe tener presente que: a) El programa ocupa memoria, por consiguiente un programa reducido, sin comentarios, con varias instrucciones por línea..., beneficiará la disponibilidad de memoria para el registro de datos, b) Las variables de cadena (\$) ocupan 3 bytes por cada variable usada, más un byte por cada caracter almacenados en ellas; las variables de números enteros (%) ocupan 2 bytes por dato y las variables de doble precisión o de número reales ocupan 5 bytes por dato, y c) los arreglos dimensionales usan 5 bytes para la declaración del arreglo y 2 bytes por cada dimensión del arreglo declarada.

La selección de las variables y tipo de variables a emplear para el registro de respuestas nuevamente dependerá de las necesidades é intereses del investigador. Este aspecto es crucial para el uso de la memoria disponible.

En el ejemplo de la rutina 3 se están registrando 180

eventos en cada uno de los 50 ensayos para ambas cajas experimentales. Los 180 eventos se suman en una variable de cadena (líneas 255 y 256) obteniendo al final de la sesión 50 variables de cadena por sujeto (una para cada ensayo) en que cada una de ellas (R1\$(NE) y R2\$(NE)) está compuesta por 180 caracteres los que representan en forma secuencial a los 180 eventos de cada ensayo. Cada respuesta está simbolizada por una letra. De esta manera se registran 18,000 eventos con un consumo bajo de memoria.

Otro modo de registro de respuestas, en donde interesa tanto el tipo de respuesta ejecutada como el tiempo en que ella ocurrió se presenta en la rutina siguiente. En este caso interesa conocer el tiempo transcurrido desde el inicio del ensayo hasta la ejecución de la respuesta. Es un programa de intervalo fijo de 60 segundos. Se emplea un arreglo de dos dimensiones en donde la primera dimensión registra el tipo de respuesta y la segunda el tiempo en que ella ocurrió.

#### Rutina 4: Lectura de tiempo de ejecución.

Las respuestas de la caja 1 se registran en R1%(N1) y el tiempo en que ella ocurrió se registra en R1(N1). Para la caja 2 las respuestas quedan registradas en R2% y el tiempo en R2(N2). El control del número de ensayos se efectúa con los contadores N1 y N2. El control del tiempo se verá en el próximo capítulo.

## C A P I T U L O    C U A T R O

### ALGUNOS ALGORITMOS DE PROGRAMACION.

En el capítulo 3 se presentaron los aspectos necesarios para programar las salidas de la interface de tal modo que fuera posible controlar la presentación de estímulos en una caja de Skinner u otro espacio experimental mediante una microcomputadora C-64. De igual manera se presentaron los aspectos requeridos para poder detectar y registrar la ocurrencia de respuestas en uno o más espacios experimentales.

En este capítulo se presentan algunos ejemplos de algoritmos de uso frecuente en el Análisis Experimental de la Conducta y al final se intenta proporcionar un procedimiento para programar los eventos de un experimento utilizando los ejemplos aquí proporcionados. En cada caso particular será necesario efectuar las modificaciones que se deriven de las características del experimento a controlar. En todo caso, las modificaciones serán pocas y de fácil realización por cualquier investigador que tenga conocimientos elementales del lenguaje BASIC.

#### a) Rutina de reforzamiento:

La presentación de reforzamiento es la simple presentación de un estímulo y por tanto solo requiere de la

activación de la línea de salida a la que esté conectada el dispositivo de entrega de reforzadores. Para ello se debe habilitar el puerto de salida correspondiente (línea 510) y poner en "1" los bits del puerto del usuario de aquellos estímulos que estarán activos durante el período de reforzamiento (línea 530). Nótese que los estímulos que participan en el período de reforzamiento aparecen conformando una suma luego de haberse habilitado el byte del puerto del usuario (POKE SS).

#### Rutina 5: Reforzamiento por 4 segundos.

En esta rutina se contempla una presentación del reforzamiento durante 4 segundos, tal como se aprecia en la línea 550 (ver más adelante el manejo del tiempo) y se supone que se está operando un dispensador de comida para palomas que requiere de un suministro continuo de energía eléctrica durante todo el intervalo que esté activo.

En la rutina 5.1 se exhibe una variante en las líneas 535 y 536 en donde se activa por un período más breve el dispositivo de entrega de alimento (dispensador de pellets para ratas), puesto que éstos actúan con un impulso de corriente eléctrica lo que equivale a un período discreto de suministro de energía.

#### Rutina 5.1: Reforzamientos por pulsos.

Nótese que en la línea 545 se está efectuando una lectura de los puertos de entrada para detectar respuestas, puesto que mientras una caja está en período de reforzamiento la otra caja sigue corriendo su experimento en forma independiente.

b) Rutina de grabado de datos:

Normalmente todos los datos generados en una sesión pueden ser mantenidos en la memoria de la microcomputadora de tal modo que puedan grabarse en diskete o casete hasta que haya terminado la sesión. Esto es aconsejable puesto que al transferir los datos al disco o casete durante el transcurso de la sesión se producen dos inconvenientes: 1) el proceso de transferencia de datos es lento y durante ese período no se puede efectuar ningún control o proceso en paralelo y 2) durante ese período el reloj de la microcomputadora se detiene pudiendo afectar a algún control que requiera el procedimiento experimental. En la rutina 6 se presenta una forma de traspasar los datos a la unidad de disco.

Rutina 6: Grabado de datos.

En esta rutina se grabaron las respuestas (R1\$(NE) y R2\$(NE)) registradas en el ejemplo proporcionado en la rutina 3. En la rutina siguiente se exhibe una variante del caso anterior en donde se graban los datos generados en la

rutina 4.

#### Rutina 6.1: Grabado de datos.

Nótese que se están grabando datos que se encuentran registrados en variables dimensionales.

#### c) Manejo del tiempo:

La C-64 tiene un reloj de tiempo real que mantiene un ciclo de 24 horas en horas, minutos y segundos. Este reloj se inicializa al encender el sistema. Existen dos variables de uso reservado para el sistema, las que contienen los datos relacionados con el manejo del tiempo por parte de la C-64. Estas variables son TI\$ y TI. La primera proporciona la hora en un formato de horas, minutos y segundo ("hhmmss") y la segunda proporciona el tiempo en unidades llamadas "jiffies" equivalentes a 1/60 de segundo (1 segundo = 60 jiffies; 1 minuto = 3,600 jiffies; 1 hora = 216,000 jiffies). La variable TI se incrementa automáticamente cada 1/60 de segundo y TI\$ es una variable de cadena que se genera a partir de TI, es decir convierte los jiffies a un formato de horas, minutos y segundos.

Para efectos de los controles necesarios de efectuar en una sesión es necesario 1) inicializar la hora en ceros al momento de comenzar la sesión, 2) utilizar una variable cualquiera (TA) para registrar el inicio de un período determinado que se quiera controlar (duración de un ensayo,

de presentación de un estímulo...), 3) utilizar otra variable (TB) para registrar el tiempo actual del período a controlar, y 4) comparar si ya ha transcurrido el intervalo de tiempo a controlar. Estos pasos se utilizan frecuentemente y aquí expondremos de manera generalizada su utilización. Las aplicaciones se pueden ver en casi todas las rutinas de ejemplo de este capítulo.

1) Inicialización de la hora: Esta instrucción debe hallarse al inicio de la sesión y actualiza la variable TI\$ en cero horas, minutos y segundos.

```
TI$="000000"
```

Nótese que apenas el programa llegue a esta instrucción el reloj se actualiza a 0 horas y a partir de ese instante la variable TI se irá incrementando constantemente cada 1/60 de segundo. En el transcurso de la sesión no se debe volver a inicializar la variable TI\$ puesto que todo volvería a 0.

2) Registro del tiempo de inicio de un período a controlar: A una variable cualquiera (TA) se debe asignar el período de inicio de un período determinado. Esta instrucción debe encontrarse inmediatamente al inicio del período.

```
TA=TI
```

La variable TA asume el valor de TI en el momento de

inicio del período a controlar.

3) Registro del tiempo actual del proceso: En otra variable arbitraria (TB) se puede registrar el tiempo actual de un proceso determinado.

TB=TI

La variable TB asume el valor de TI en el momento en que se efectúe el control. Esta variable deberá ser actualizada frecuentemente con el objeto de no dejar pasar mucho tiempo sin efectuar controles.

4) Comparación del tiempo transcurrido: Supóngase que se está controlando la duración de un período de 4 segundos, lo que equivale a 240 jiffies. Nótese que toda comparación se debe efectuar en unidades jiffies. Una instrucción como la del ejemplo siguiente comparará si han transcurrido mas de 240 jiffies. Si aún no han transcurrido los 240 jiffies se vuelve a leer TI y su estado se registrará en TB.

IF TB =< TA+240 THEN GOTO leer TB

Si simultáneamente se están controlando varios períodos (tiempo de sesión, tiempo de ensayos, tiempo de reforzamiento...) será necesario utilizar diferentes nomenclaturas para TA y TB con el objeto de no crear confusiones. Es preciso tener en cuenta que el reloj de la C-64 se detiene cuando se accesa algun dispositivo

periférico (unidad de disco, casetera, impresora).

d) Manejo de probabilidades:

Es común en el Análisis Experimental de la Conducta utilizar valores aleatorios para determinar la presentación de algunos estímulos o para definir la duración de la presentación de ciertos estímulos, etc. La C-64 trae incorporada una función (RND) que genera números aleatorios.

La función RND crea un número real comprendido entre el 0.0 y el 1.0. La microcomputadora genera una secuencia de números aleatorios al efectuar cálculos a partir de un número dado, al que se denomina "semilla". Los números surgen de un cálculo efectuado a partir de las cifras provistas por tres relojes del sistema. Como el algoritmo de cálculo siempre es el mismo, en realidad los números son pseudoaleatorios los que, sin embargo, son aceptados como aleatorios. A continuación se presentan diversas aplicaciones de esta función.

```
409 REM ** IMPRIME 20 NUMEROS ALEATORIOS DEL 0 AL 49 **
410 FOR I=1TO20:REM * EFECTUA 20 VECES INSTRUCCION 420 *
420 PRINT INT(RND(0)*50)
430 NEXT I
```

El siguiente ejemplo asigna a la variable X un valor aleatorio comprendido entre 1 y 100

```
420 X= INT(RND(1)*100)+1
```

El ejemplo siguiente asigna a X un valor aleatorio comprendido en el rango de 100 a 249.

```
420 X= INT(RND(1)*150)+100
```

En general se puede usar el siguiente procedimiento para generar números aleatorios en un rango cuyo límite inferior es LI y su límite superior está representado por la variable LS.

```
420 X= INT(RND(1)*(LS-LI)+LI)
```

El uso de esta función es muy variado y puede ser requerido en diferentes partes del programa, dependiendo todo ello de las necesidades del investigador.

e) Programas de razón:

En el Análisis Experimental de la Conducta los programas de razón están determinados por una cantidad de repuestas ejecutadas para poder acceder al período de reforzamiento. Estos pueden ser de reforzamiento continuo o de reforzamiento intermitente. El primer caso estipula que cada respuesta ejecutada en un operando determinado, y bajo ciertas condiciones, ocasionará la presentación de reforzamiento. Esto implica que si en la rutina de lectura de respuesta se detecta la ocurrencia de la respuesta en el operando especificado debe alterarse la secuencia del

programa para que éste ejecute la rutina de reforzamiento. Esto se traduciría en una instrucción del siguiente tipo en la rutina 3. La respuesta en cuestión sería la ejecutada en el operando conectado al bit 1 de la caja 1 y en el bit 0 de la caja 2.

```
250 IF (J AND 2)=0 THEN GOSUB ** RUTINA DE REF. CAJA 1 **  
260 IF (K AND 1)=0 THEN GOSUB ** RUTINA DE REF. CAJA 2 **
```

Entre los programas de reforzamiento intermitente hay que considerar los programas de razón fija y de razón variable. En los de razón fija es preciso tener en cuenta tres elementos: 1) el valor de la razón fija, 2) un contador de respuestas que lleve el control de las clases de respuestas ejecutadas en cada ensayo, y 3) una instancia de comparación del total de determinada clase de respuestas ejecutadas en un ensayo con el valor de la razón fija. En la rutina 7 se presentan estas tres etapas. La primera contempla la petición del valor de la razón fija por pantalla en una etapa de inicialización del programa computacional.

#### Rutina 7: Razón fija.

En los renglones 250 y 260 se incrementan los contadores RA y RB siempre y cuando se haya ejecutado la respuesta prevista en la caja 1 (contador RA) o en la caja 2 (contador RB). En los renglones 251 y 261 se efectúa la

comparación de cada contador con el valor de la razón fija. En el caso de que se haya cumplido la razón esperada el programa es enviado a una rutina de reforzamiento. La rutina de reforzamiento de un programa de razón fija debe incluir dos instrucciones adicionales a las mostradas en los ejemplos de reforzamiento anteriores. Estas son 1) una instrucción que habilite la lectura del teclado y 2) una instrucción que reinicialice los contadores de respuestas. Estas dos instrucciones podrían tomar la siguiente forma:

```
502 POKE 56322,255: REM ** REHABILITA TECLADO **  
503 RA=0 : REM ** REINICIALIZA CONTADOR DE CAJA 1 **  
510 REM** ENTRA RUTINA DE REF. CAJA 1 **
```

En el caso de los programas de razón variable la cantidad de respuestas a ejecutar por ensayo para acceder al reforzamiento varía de ensayo a ensayo. Aquí se pueden manejar varias opciones para establecer los valores de la razón para cada ensayo. Entre ellas se pueden mencionar 1) el ingreso de los valores de la razón para cada ensayo por pantalla y 2) la generación de los valores para cada ensayo por medio de la función aleatoria RND. El primer caso se ejemplifica en la Rutina 7.1 e implica la captura de los datos antes de iniciar la sesión, y la segunda alternativa se muestra en la rutina 7.2 en donde es preciso agregar una instrucción que genere el valor requerido para cada ensayo.

Rutina 7.1: Razón variable.

En este ejemplo se solicitan los valores de la razón para cada ensayo. Antes se ha solicitado el número de ensayos a correr en la sesión (renglón 50) y este valor queda consignado en la variable NE. En el renglón 51 se inicializa la variable E1 y E2 que fungirán como contadores del número de ensayos para cada caja. En los renglones 60 al 90 se solicitan los valores para cada ensayo, los que quedarán registrados en la variable VR(I). La lectura de respuestas, incremento de contadores y comparación del número de respuestas correctas ejecutadas en cada ensayo es similar al ejemplo anterior con la única diferencia que la comparación se efectúa contra un valor de razón diferente para cada ensayo (VR(E1) o VR(E2)).

Para estos casos la rutina de reforzamiento debe contar con las mismas instrucciones mostradas para la rutina 7 más dos adicionales. Una simplemente es un incremento al contador del número de ensayos (E1 o E2) y la otra es una comparación para controlar si se ha cumplido el número total de ensayos programados (renglón 590). Si ya se cumplieron los ensayos previstos el programa es llevado a una rutina de finalización de la sesión.

```
502 POKE 56322,255
503 RA=0
504 E1=E1+1
510 REM ** ENTRA RUTINA DE REF. CAJA 1 **
590 IF E1=NE+1 THEN GOTO ** FIN DE LA SESION**
```

Si se quiere emplear una función RND para calcular el valor de la razón para cada ensayo la rutina podría ser como el ejemplo siguiente:

Rutina 7.2: Razon Variable/ RND.

Para este ejemplo la rutina de reforzamiento debe contener las siguientes instrucciones:

```
502 POKE 56322,255
503 RA=0
510 REM ** ENTRA RUTINA DE REF. CAJA 1 **
590 GOSUB 200
600 RETURN
```

La línea 590 envía el programa a la instrucción 200 la que calcula el nuevo valor de X para el próximo ensayo de la caja 1. Para la caja 2 el valor de la nueva razón se calcula en el renglón 201.

Estos ejemplos son demostrativos de un modo de efectuar los controles del número de ensayos de una sesión, del número de respuestas ejecutadas por ensayo y de los valores asumidos por cada razón en cada ensayo. Las posibilidades de efectuar estos pasos por medio de otros algoritmos son muchas y todo dependerá de los intereses del investigador y de las habilidades del programador.

f) Programas de intervalo:

Los programas de intervalo están definidos por el cumplimiento de dos requisitos: a) el transcurrir de cierto

tiempo y b) la ocurrencia de cierta respuesta después de ese tiempo. Al cumplirse estas dos condiciones se tiene acceso al reforzamiento. Dentro de los programas de intervalo se encuentran los de intervalo fijo y los de intervalo variable. En ambas modalidades lo crítico es el control del tiempo. Ejemplos de intervalo fijo se han presentado en la rutina 3 y 4. Un ejemplo general podría ser el presentado a continuación (rutina 8), en donde se solicita el valor del intervalo en la etapa de inicialización del programa computacional.

#### Rutina 8: Intervalo fijo.

En este caso el control del tiempo y la rutina de reforzamiento es simultánea e idéntica para ambas cajas. La rutina de reforzamiento debe terminar con una instrucción que regrese el programa a la línea 100. El control del número de ensayos puede encontrarse en la rutina de reforzamiento.

Para el caso de un programa de intervalo variable el procedimiento de asignación de valores para cada ensayo es similar a lo visto en los programas de razón variable. La rutina 8.1 muestra la asignación aleatoria del valor del intervalo para cada ensayo de cada caja.

#### Rutina 8.1: Intervalo variable /RND.

En el ejemplo anterior hay varios aspectos que comentar. Uno de ellos es que cada caja está funcionando con un valor diferente de intervalo para cada ensayo y por tanto su funcionamiento sera asincrónico. Esto implica que se deben efectuar controles independientes de la duración del ensayo para cada caja (caja 1 : renglones 110 al 140 y caja 2 : renglones 150 al 180). Otro aspecto es que en todo momento se debe estar verificando el transcurso del tiempo (TB y TD) y efectuando lecturas de los puertos de entrada. Eso significa que si al controlar el tiempo de la caja 1 no se ha cumplido el requisito del intervalo, entonces se irá a leer el puerto de entrada de la caja 1 (renglón 130). Esta lectura de puerto debe concluir con una transferencia del programa a la línea 160, la que actualiza el estado del tiempo (TD) de la caja 2. Si aún no se cumple el requisito del intervalo se irá a leer el puerto 2 de entrada y esta rutina de lectura debe terminar con una transferencia a la línea 120. Nótese que el programa inicializó las variables TA y TC antes de generar por primera vez los valores de X y Y y por tanto los valores de TA y TC sólo serán válidos para el primer ensayo de cada caja. En lo sucesivo, cada rutina de reforzamiento debe contemplar la actualización de los valores de TA y TC. Por tanto las rutinas de reforzamiento de estos programas deben incluir la actualización de TA para la caja 1 y de TC para la caja 2. Además deben considerar la generación del valor de X para la caja 1 y de Y para la caja 2, los que serán los valores de los intervalos para el

próximo ensayo.

Otro aspecto a considerarse en la rutina de reforzamiento de los programas de intervalo variable es la lectura del puerto de entrada de la otra caja durante el período de reforzamiento, tal como se aprecia en el ejemplo proporcionado por la rutina 5, línea 545.

g) Captura de parámetros por pantalla:

Es aconsejable que ciertos datos, tales como fecha de la sesión, nombre del sujeto, fase del experimento, valores de razón o intervalo, nombre del archivo a generar, u otros de índole similar, cuya característica común es que son datos que pueden variar cada vez que se corra el programa computacional y por tanto es preferible que éstos sean proporcionados por el investigador al iniciar el programa. Por consiguiente este detalle debe estar contemplado en alguna rutina del programa, de tal modo que al correr el programa éste solicite al usuario que ingrese los datos requeridos, definiendo así los parámetros particulares para cada sesión.

Un ejemplo general se proporciona en la rutina 9. Los valores ingresados por el investigador quedan asignados a las variables que aparecen al final de cada sentencia y cuyo uso en el transcurso de la sesión puede ser muy variado.

Rutina 9: Datos de inicialización.

Los datos aquí solicitados son sólo ejemplos y su uso puede ser muy variado. Otros ejemplos se pueden observar en los programas incluidos en el apéndice de esta tesis. En ellos se puede apreciar la interacción del investigador con la microcomputadora proporcionándole datos que serán utilizados en diferentes procesos, ya sea antes de iniciar la sesión como al término de la misma. En la rutina 9.1 se presenta un ejemplo de interacción por medio de la pantalla y el teclado al finalizar una sesión.

#### Rutina 9.1: Fin de la sesión.

En este ejemplo se presentan opciones al usuario para que ésta decida si quiere grabar los datos, o imprimirlos o visualizarlos por pantalla. El usuario podrá elegir una o más de una de las opciones posibles. El ejemplo sólo considera los datos generados en la caja 1. Para trabajar con los datos de la caja 2 habría que modificar solamente las variables pertinentes.

#### h) Análisis de datos:

Al tener todos los datos relevantes de cada sesión almacenados en disketes o casetes es posible volver a ellos las veces que sea necesario para aplicarle una u otra forma de análisis. El tipo de análisis a efectuar dependerá de los intereses del investigador.

Usualmente se utiliza algún tipo de análisis de datos

ya disponibles en un paquete computacional. En estos casos lo importante es generar los archivos adecuados para que éstos puedan ser leídos en forma correcta por el programa computacional a emplear. Considerando que la norma en esta área está siendo impuesta por los paquetes usados por computadoras personales compatibles con IBM, en la rutina 10 se presenta un programa de conversión de los archivos secuenciales de la C-64 a una forma compatible con la procesada por la PC IBM.

#### Rutina 10: Conversión de datos de C-64 a PC IBM.

Además de estos análisis estadísticos es posible que el investigador quiera efectuar algún ordenamiento particular de datos o un análisis simple de los mismos. Esto es relativamente fácil realizarlo mediante la elaboración de un programa en Basic. En el apéndice de "programas" se presentan dos programas de análisis particulares de datos.

#### i) Sugerencias para elaborar un programa computacional:

El seguimiento de instrucciones por parte de una computadora siempre es secuencial, a no ser que haya una instrucción especial que indique alterar la secuencia. Estas instrucciones en BASIC son GOTO, GOSUB, ON X GOSUB.

En todo programa hay algunos pasos que pueden ser cambiados y otros no, los que son cruciales para una correcta ejecución de lo requerido.

En términos generales se podría decir que un programa computacional es la traducción de un procedimiento determinado a una serie de instrucciones comprensibles para el sistema computacional. La traducción del procedimiento a un lenguaje computacional no reviste complejidad alguna si el procedimiento está bien formulado y si se conoce la sintaxis del lenguaje a emplear. Si damos por hecho que la sintaxis del BASIC es una de las más simples de dominar, entonces en donde se deben contemplar todas las contingencias posibles es en la formulación del procedimiento.

Se debe tener presente lo siguiente antes de formular el procedimiento: 1) el procedimiento está compuesto por un número finito de etapas que se deben ejecutar en un orden específico, 2) si el problema a resolver tiene solución, entonces el procedimiento debe encontrar esa solución. Sin embargo, si el problema no conduce a una solución, entonces el procedimiento puede o bien reconocer esa situación o continuar indefinidamente, y 3) el procedimiento debe ser compatible con el medio utilizado para llevarlo a cabo. Esto quiere decir que se debe expresar en un lenguaje que pueda ser interpretado por el medio a emplear y en función de las operaciones que éste puede ejecutar.

Teniendo en consideración lo anterior es aconsejable plantear por escrito las diferentes etapas y pasos que debe realizar el procedimiento antes de traducirlo a un lenguaje computacional. Al contemplar las posibles eventualidades y

operaciones que debe cubrir el proceso se tendrá claro que es lo que se quiere hacer por medio de la computadora. Una vez logrado esto, es necesario revisar si todas las operaciones contempladas en el procedimiento son compatibles con la capacidad de la computadora. Si la respuesta es positiva, entonces la traducción del procedimiento a un lenguaje computacional es simple y se puede efectuar sin mayores dificultades adaptando las rutinas proporcionadas en los diversos ejemplos anteriores.

En el apéndice de las "rutinas" se presenta un procedimiento general de una sesión experimental. Este algoritmo general debe adaptarse a las especificaciones propias de un experimento particular.

#### Algoritmo general.

A continuación se presenta el procedimiento escrito de un programa de entrenamiento en comedero en una caja para palomas, trabajando con dos comederos operativos y con focos en cada uno de los comederos.

#### Procedimiento de entrenamiento en comedero.

En la rutina 11 se podrá encontrar la traducción de este procedimiento en BASIC.

#### Rutina 11: Entrenamiento en comedero.

Una observación final se relaciona con la aplicación de las rutinas proporcionadas como ejemplos. Al utilizar algunas de éstas se debe prestar especial atención a la correcta concatenación de las numeraciones de las instrucciones y de reemplazar por las correspondientes sintaxis apropiadas aquellas instrucciones que contiene indicaciones precedidas de asteriscos.

## C A P I T U L O   C I N C O

### REPORTE EXPERIMENTAL

El presente capítulo da cuenta de un experimento en desarrollo el que utiliza la interface descrita en capítulos anteriores. Aquí se reporta una breve introducción a la temática del experimento, su método y el tipo de análisis de datos a emplear. Dado que los datos aún están generándose no se presentará una discusión de los resultados.

#### INTRODUCCION:

En los programas de reforzamiento de Tiempo Fijo (TF), en los que se presenta comida cada cierto intervalo fijo de tiempo en forma independiente de la ejecución del sujeto, las ratas muestran una tendencia creciente a visitar el área de comedero, olfatearla e incluso roerla a medida que se aproxima el momento de entrega de alimento (Davies y Hubbard, 1972; Staddon y Ayres, 1975; Ribes y Chávez, 1988). En los programas de reforzamiento de Intervalo Fijo (IF), en los que la entrega de comida es dependiente de la ocurrencia de una respuesta particular, es común observar que la respuesta terminal escogida por las ratas sea el presionar una palanca (Ferster y Skinner, 1957). Son variadas las actividades interinas que pueden ocurrir en programas de TF y IF, y sus características dependerán en gran medida de las

oportunidades disponibles en la situación experimental y de las especies usadas; algunos ejemplos son el beber, agresión, acicalamiento y otras actividades (ver revisiones de Falk, 1971; Staddon, 1977).

Existen varios estudios que analizan el efecto de drogas sobre la organización conductual en programas de IF y TF (Sanger y Blackman, 1976; Nieto, Makhlouf y Rodríguez, 1979; Nieto y Posadas-Andrews, 1984). El presente estudio se diseñó para evaluar el efecto de drogas en la organización conductual de ratas en un programa de TF en una situación experimental que permite la manipulación de tres operandos.

#### METODO:

a) Sujetos: Se utilizan 8 ratas albinas, machos, ingenuas. Cuatro de ellas privadas de alimento y mantenidas al 80% de su peso ad libitum. Las otras cuatro tienen acceso libre a la comida. Todas cuentan con caja habitación individual y agua disponible continuamente.

b) Aparatos: Se utilizan 2 cajas idénticas manufacturadas por BRS Foringer cuyas dimensiones son 24 cm. de ancho, 33 cm. de alto y 31 cm. de largo. El área de circulación de la rata estaba conformada por una parrilla de 14 barras conectadas a un dispensador de choques y ubicada a 9 cm. del piso. El panel de trabajo y la pared opuesta a éste son de aluminio y todas las demás, incluyendo el techo, son de acrílico translúcido. Ambas cajas están aisladas acústica y

visualmente por recámaras que contienen un ventilador que funciona continuamente. El panel de trabajo es idéntico para ambas cajas y está compuesto por tres operandos. Al centro y al nivel del área de circulación de la rata se encuentra una puerta batiente de acrílico de 5.5 x 6.5 cm. conectada a un interruptor, la que permite registrar los movimientos provocados por el sujeto al acceder al área de comedero. Un dispensador de pellets (Campden Instruments Ltd., 45 mg., dieta completa) hace llegar en forma automática las pellets al área de comedero. A un costado de la puerta batiente se encuentra una palanca a 8 cm. del área de trabajo de la rata, forrada con suela sintética (ésto con el objeto de detectar posibles conductas de roer). Al otro costado de la puerta batiente y a 8 cm. del área de trabajo se ubica un tubo de acero que provee de agua al efectuar un contacto con él. Los contactos del sujeto con el bebedero son detectados por un circuito amplificador inversor descrito en el capítulo tres de esta tesis. Las cajas cuentan con iluminación general provista por dos focos localizados en la parte superior de la pared opuesta al panel de trabajo. El control de eventos, registro de respuestas, almacenamiento y análisis de datos se realizan con una microcomputadora Commodore 64 equipada con la interface objeto de esta tesis. Los programas computacionales que sirven para correr el experimento y los de análisis de datos se presentan en un apéndice de este trabajo.

c) Procedimiento: A los cuatro sujetos privados se les entrega pellets al cabo de 60 segundos, independiente de su ejecución. Previamente, estos sujetos fueron entrenados a comer en el area de comedero. La entrega de pellets es acompañada por iluminación del comedero durante cuatro segundos al término de los cuales se enciende la luz general la que permanece en esa condición durante los 60 segundos de cada ciclo. A los otros sujetos se les administra un choque eléctrico de 0.5 amperes durante 0.5 segundo al cabo de 60 segundos, independiente de la ejecución del sujeto. El choque no es precedido por ningún tipo de señalamiento. Ambos grupos corren sesiones diarias 6 veces a la semana. Cada sesión está compuesta por 50 ciclos. Para los dos grupos se efectuan registros de su actividad cada tercio de segundo, obteniéndose de esta manera 180 registros por ciclo. Las cateorías conductuales que se registran son: A) respuesta en palanca, B) respuesta en bebedero, C) resp'esta en el área de comedero y D) no respuesta en ningún operando. En una fase posterior se considera la administración de drogas con el objeto de evaluar su efecto en la organización conductual de estos dos grupos de sujetos.

d) Análisis: Para cada sesión se agrupaban los datos de la siguiente manera: a) Total de cada categoría de respuesta por bins en cada ciclo (10 bins por cada ciclo), por ciclo y por sesión, b) Total de cada categoría de respuestas por bins, c) Total de tiempo dedicado a cada categoria de

respuesta por sesión. Cada dato es proporcionado en frecuencia relativa.

En todo momento es posible aplicar algún otro tipo de análisis puesto que todos los datos están disponibles para procesarlos con algún otro metodo.

Se presenta el análisis de los datos de una sesión inicial para un sujeto al que se le administra pellets y otra de un sujeto en la condición de choque electrico.

SUJETO : JN1

FECHA : 880831

TOTAL DE RESPUESTAS POR BINS

BIN1	BIN2	BIN3	BIN4	BIN5	BIN6	BIN7	BIN8	BIN9	BIN10	TOTAL
PALANCA :										
1	13	4	2	6	3	7	3	6	0	45
PUERTA :										
18	42	62	56	50	25	55	33	45	26	412
BEBEDERO:										
16	7	7	2	12	9	21	5	1	6	86
NO RESP.:										
1027	1000	989	1015	1012	1043	997	1039	1028	1048	10198

TOTAL DE RESPUESTAS POR SESION

	TOTAL RESPUESTAS	TIEMPO SEG.	FRECUENCIA RELATIVA
PALANCA :	45	15	0
PUERTA :	412	137	3
BEBEDERO:	86	28	0
NO RESP.:	10198	3399	94

CONDICION: REFORZAMIENTO CON PELLETS.

SUJETO : JN2

FECHA : 880830

TOTAL DE RESPUESTAS POR BINS

BIN1	BIN2	BIN3	BIN4	BIN5	BIN6	BIN7	BIN8	BIN9	BIN10	TOTAL
PALANCA :										
0	0	0	0	0	0	0	0	0	0	0
PUERTA :										
206	213	183	191	212	172	203	198	179	207	1964
BEBEDERO:										
38	39	39	43	53	18	20	47	68	44	409
NO RESP.:										
818	810	840	841	815	890	857	835	833	829	8368

TOTAL DE RESPUESTAS POR SESION

	TOTAL RESPUESTAS	TIEMPO SEG.	FRECUENCIA RELATIVA
PALANCA :	0	0	0
PUERTA :	1964	654	18
BEBEDERO:	409	136	3
NO RESP.:	8368	2789	77

CONDICION : REFORZAMIENTO CON PELLETS

SUJETO : JN4

FECHA : 880830

TOTAL DE RESPUESTAS POR BINS

BIN1	BIN2	BIN3	BIN4	BIN5	BIN6	BIN7	BIN8	BIN9	BIN10	TOTAL
PALANCA :										
0	2	0	0	0	0	0	0	0	0	2
PUERTA :										
4	0	0	2	5	7	5	2	2	2	29
BEBEDERO:										
2	1	1	5	3	0	0	0	2	1	15
NO RESP.:										
1056	1059	1061	1068	1072	1073	1075	1078	1076	1077	10695

TOTAL DE RESPUESTAS POR SESION

	TOTAL RESPUESTAS	TIEMPO SEG.	FRECUENCIA RELATIVA
PALANCA :	2	0	0
PUERTA :	29	9	0
BEBEDERO:	15	5	0
NO RESP.:	10695	3565	99

CONDICION: REFORZAMIENTO CON PEELETS

SUJETO : CH2

FECHA : 880830

TOTAL DE RESPUESTAS POR BINS

BIN1	BIN2	BIN3	BIN4	BIN5	BIN6	BIN7	BIN8	BIN9	BIN10	TOTAL
PALANCA :										
0	0	0	0	0	0	0	0	0	0	0
PUERTA :										
0	0	0	0	0	0	0	0	0	0	0
BEBEDERO:										
19	15	18	18	19	20	22	23	20	26	200
NO RESP.:										
1043	1047	1044	1057	1061	1060	1058	1057	1060	1054	10541

TOTAL DE RESPUESTAS POR SESION

	TOTAL RESPUESTAS	TIEMPO SEG.	FRECUENCIA RELATIVA
PALANCA :	0	0	0
PUERTA :	0	0	0
BEBEDERO:	200	66	1
NO RESP.:	10541	3513	98

CONDICION : CHOQUE ELECTRICO



Figura 1.1: Diagramas de un foco.

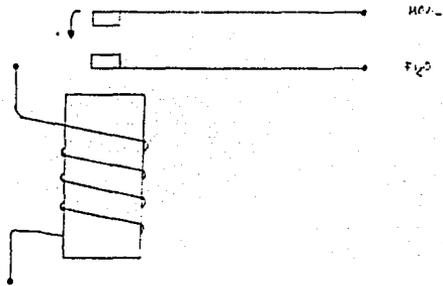


Figura 1.2: Diagrama de un relevador.

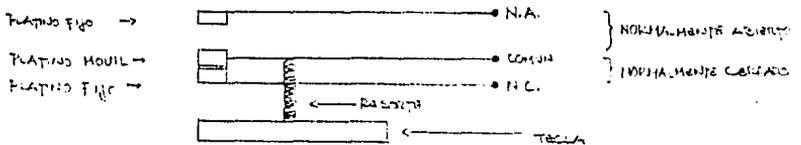


Figura 1.3: Interruptor en condición "normalmente cerrado".

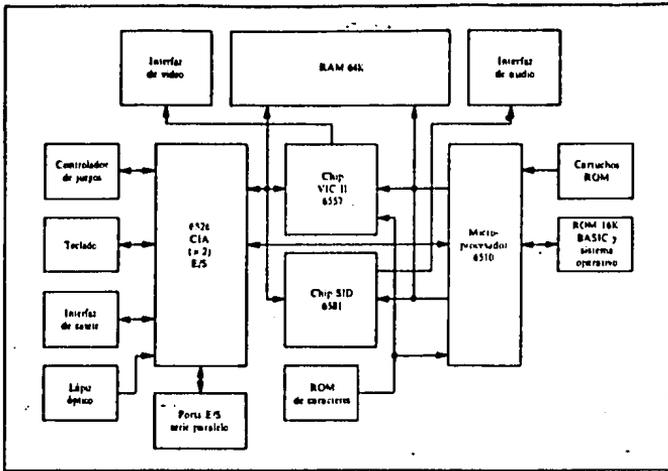


Figura 1.4: Diagrama de bloques de la C-64

		B	I	T	S				
		7	6	5	4	3	2	1	0
BYTE	56320,15	0	0	0	0	1	1	1	1
BYTE	56321,255	1	1	1	1	1	1	1	1
BYTE	56322,0	0	0	0	0	0	0	0	0
BYTE	56323,31	0	0	0	1	1	1	1	1

Figura 1.8: Ejemplos de valores decimales de diferentes bytes.

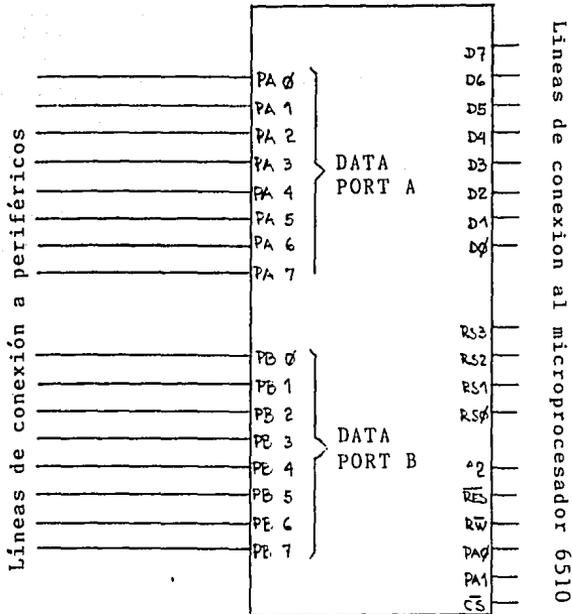


Figura 2.1: Diagrama esquemático de los Data Port A y Data Port B del C.I.A. 6526

Figura 2.3:  
Matriz de 8 x 8  
bits del teclado.

		READ PORT B (56321, SDC01)							
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WRITE TO PORT A 56320/SDC00	Bit 7	STOP	Q	G	SPACE	2	CTRL	←	1
	Bit 6	/	↑	=	RIGHT SHIFT	HOME	;	*	£
	Bit 5	/	@	:	.	-	L	P	+
	Bit 4	N	,O	K	M	0	J	I	9
	Bit 3	V	U	H	B	8	G	Y	7
	Bit 2	X	T	F	C	6	D	R	5
	Bit 1	LEFT SHIFT	E	S	Z	4	A	W	3
	Bit 0	CRSR DOWN	.5	.3	.1	.7	CRSR RIGHT	RETURN	DELETE

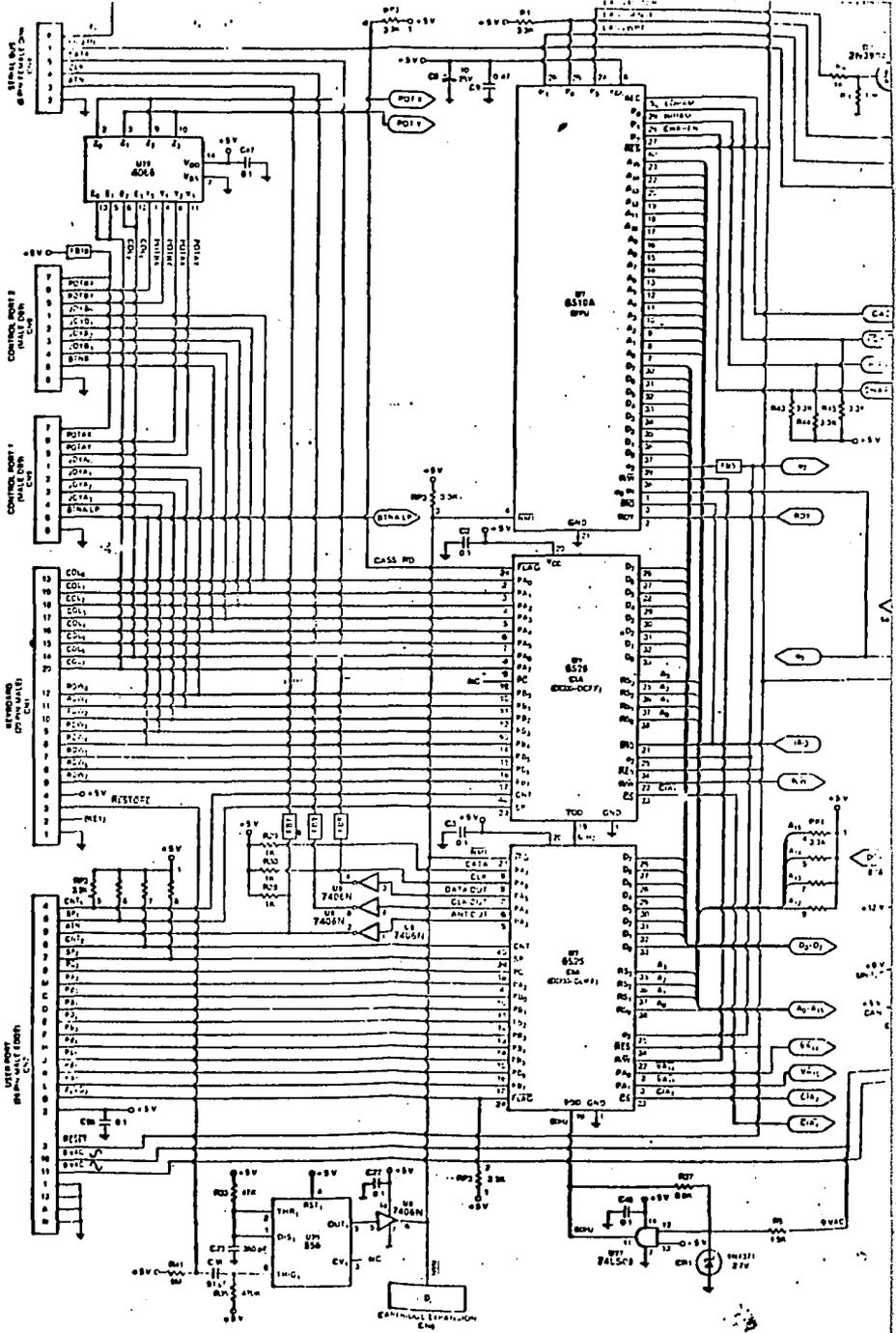


FIGURA 2.2: DIAGRAMA PARCIAL DE LOS CIRCUITOS DE LA C-64

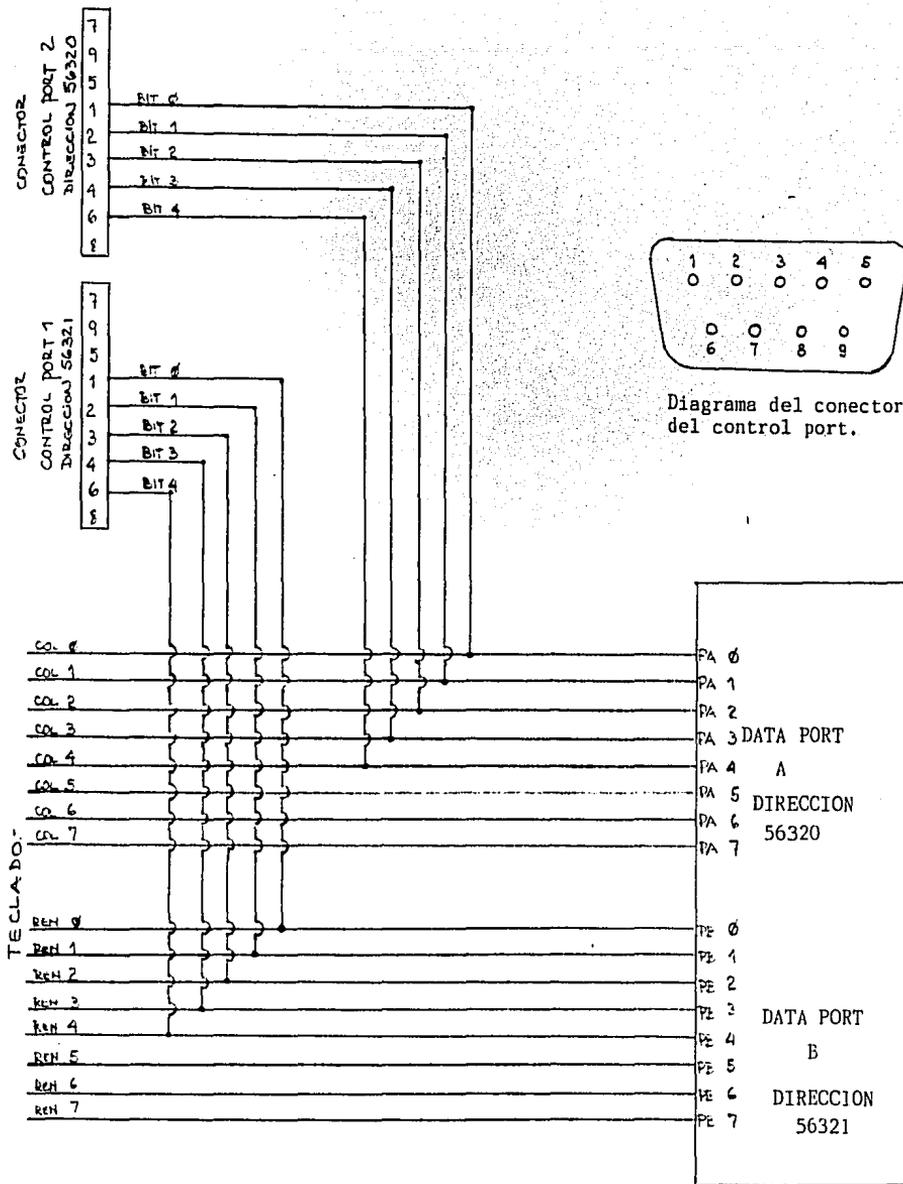
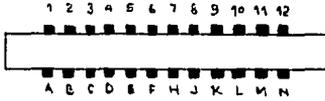


Figura 2.4: Conexiones del CIA 6526 #1 al teclado y al Control Port 1 y Control Port 2.



Conectores del Puerto del Usuario.

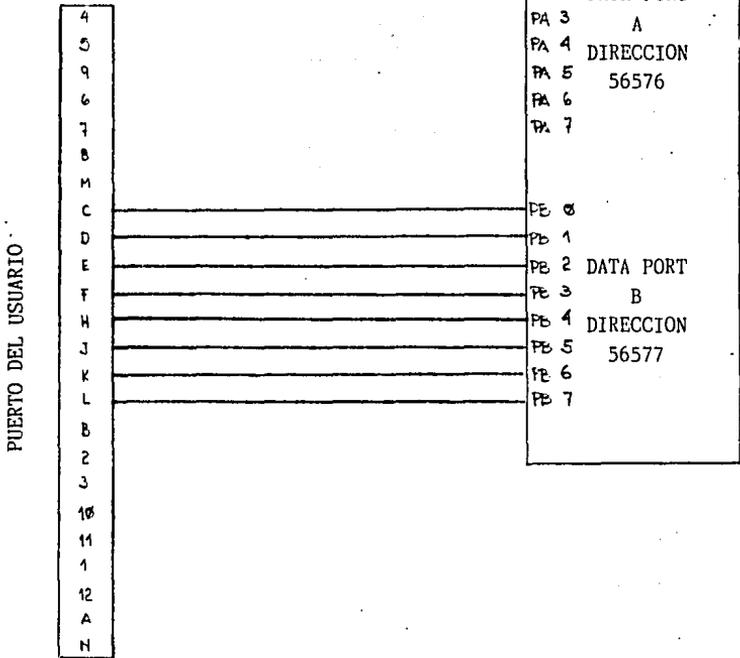


Figura 2.5: Conexiones del Dta Port B del C.I.A. 6526 #2 al Puerto del Usuario.



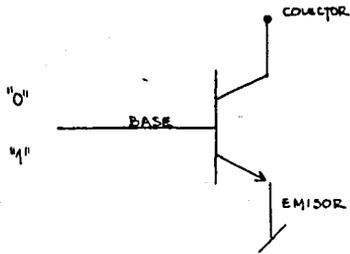


Figura 2.8: Diagrama de un transistor.

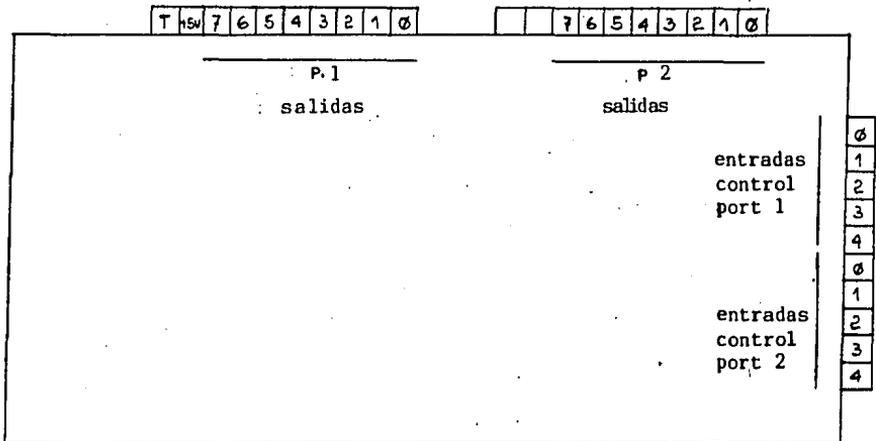


Figura 2.9: Conectores exteriores de la interface

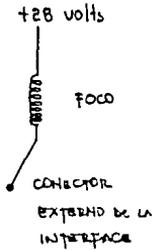


Figura 2.10: Conexión de un foco al conector de salida de la interface.

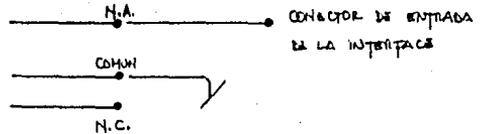


Figura 2.11: Conexión de un interruptor al conector de entrada de la interface.

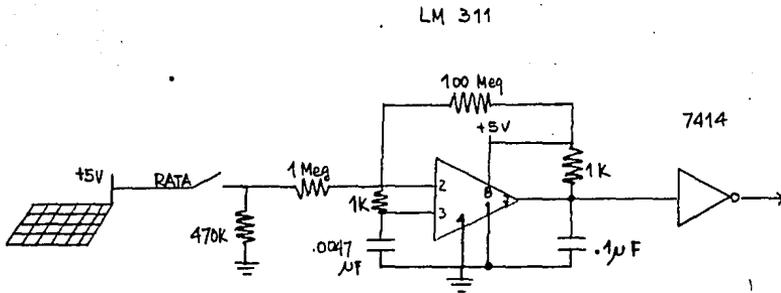


Figura 2.12: Diagrama de circuito amplificador inversor.

DATA PORT REGISTER A: Dirección de memoria 56320

- Bit 0: Lectura de la columna 0 del teclado  
Lectura del controlador de juegos 2, arriba.
- Bit 1: Lectura de la columna 1 del teclado  
Lectura del controlador de juegos 2, abajo.
- Bit 2: Lectura de la columna 2 del teclado  
Lectura del controlador de juegos 2, izquierda.
- Bit 3: Lectura de la columna 3 del teclado  
Lectura del controlador de juegos 2, derecha.
- Bit 4: Lectura de la columna 4 del teclado  
Lectura del controlador de juegos 2, disparo.
- Bit 5: Lectura de la columna 5 del teclado
- Bit 6: Lectura de la columna 6 del teclado  
Selecciona para leer paletas en Puerto A o B.
- Bit 7: Lectura de la columna 7 del teclado  
Selecciona para leer paletas en Puerto A o B.

DATA PORT REGISTER B: Dirección de memoria 56321

- Bit 0: Lectura del renglón 0 del teclado  
Lectura del controlador de juegos 1, arriba.
- Bit 1: Lectura del renglón 1 del teclado  
Lectura del controlador de juegos 1, abajo.
- Bit 2: Lectura del renglón 2 del teclado  
Lectura del controlador de juegos 1, izquierda.
- Bit 3: Lectura del renglón 3 del teclado  
Lectura del controlador de juegos 1, derecha.
- Bit 4: Lectura del renglón 4 del teclado
- Bit 5: Lectura del renglón 5 del teclado  
Lectura del controlador de juegos 1, disparo.
- Bit 6: Lectura del renglón 6 del teclado  
Envía un pulso de salida al Timer A.
- Bit 7: Lectura del renglón 7 del teclado  
Envía un pulso de salida al Timer B.

Tabla 1: Funciones de los bits del Data Port A y B del CIA  
6526 # 1

DATA DIRECTION REGISTER A: Dirección de memoria 56322

Nota: La condición original del sistema establece un "1" para cada bit.

- Bit 0: Selecciona el bit 0 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 1: Selecciona el bit 1 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 2: Selecciona el bit 2 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 3: Selecciona el bit 3 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 4: Selecciona el bit 4 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 5: Selecciona el bit 5 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 6: Selecciona el bit 6 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 7: Selecciona el bit 7 del Data Port A para entrada o salida (0=entrada, 1=salida)

DATA DIRECTION REGISTER B: Dirección de memoria 56323

Nota: La condición original del sistema establece un "0" para cada bit.

- Bit 0: Selecciona el bit 0 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 1: Selecciona el bit 1 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 2: Selecciona el bit 2 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 3: Selecciona el bit 3 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 4: Selecciona el bit 4 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 5: Selecciona el bit 5 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 6: Selecciona el bit 6 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 7: Selecciona el bit 7 del Data Port B para entrada o salida (0=entrada, 1=salida)

Tabla 2: Función control de cada bit del DDR A y B del CIA 6526 # 1.

DATA PORT REGISTER A: Dirección de Memoria 56576

- Bit 0: Selecciona bancos de memoria disponibles para gráficas en video.
- Bit 1: Selecciona bancos de memoria disponibles para gráficas en video.
- Bit 2: TXD (Transmitted data), en conexión con un cartucho RS-232.
- Bit 3: ATN , salida puerto serial.
- Bit 4: CLOCK, salida puerto serial.
- Bit 5: DATA, salida puerto serial.
- Bit 6: CLOCK, entrada puerto serial.
- Bit 7: DATA, entrada puerto serial.

DATA PORT REGISTER B: Dirección de memoria 56577

Nota: Bit 0 a Bit 7 generalmente libres y disponibles. Al conectarse con un cartucho RS-232 asumen las siguientes funciones:

- Bit 0: RXD (receive data)/Conector C del puerto del Usuario.
- Bit 1: RTS (request to send)/Conector D del puerto del Usuario.
- Bit 2: DTR (data terminal ready)/Conector E del puerto del Usuario.
- Bit 3: RI (ring indicator)/Conector F del puerto del Usuario.
- Bit 4: DCD (data carrier detect)/Conector H del puerto del Usuario.
- Bit 5: libre / Conector J del Puerto del Usuario.
- Bit 6: CTS (clear to send)/Conector K del puerto del Usuario.
- Bit 7: DSR (data set ready)/Conector L del puerto del Usuario.

Tabla 3: Funciones de los bits de los Data Port A y B del CIA 6526 # 2.

DATA DIRECTION REGISTER A: Direccion de memoria 56578

Nota: La condición original del sistema establece un "1" para todos los bits, excepto bit 6 y 7.

- Bit 0: Selecciona el bit 0 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 1: Selecciona el bit 1 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 2: Selecciona el bit 2 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 3: Selecciona el bit 3 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 4: Selecciona el bit 4 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 5: Selecciona el bit 5 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 6: Selecciona el bit 6 del Data Port A para entrada o salida (0=entrada, 1=salida)
- Bit 7: Selecciona el bit 7 del Data Port A para entrada o salida (0=entrada, 1=salida)

DATA DIRECTION REGISTER B: Direccion de memoria 56579

Nota: La condición original del sistema establece un "0" para cada bit.

- Bit 0: Selecciona el bit 0 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 1: Selecciona el bit 1 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 2: Selecciona el bit 2 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 3: Selecciona el bit 3 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 4: Selecciona el bit 4 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 5: Selecciona el bit 5 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 6: Selecciona el bit 6 del Data Port B para entrada o salida (0=entrada, 1=salida)
- Bit 7: Selecciona el bit 7 del Data Port B para entrada o salida (0=entrada, 1=salida)

Tabla 4: Función control de cada bit del DDR A y B del CIA 6526 # 1.

```
READY
1 ****** RUTINA 0 *****
2 *** RUTINA DE INICIALIZACION ***
10 POKE 56578,63:POKE 56579,255: REM ** ESTABLECE DDRS PARA DPA Y DPB **
20 PU=56576:REM ** ASIGNA A PU LA DIRECCION DEL DPA **
30 SS=56577:REM ** ASIGNA A SS LA DIRECCION DEL DPB **
40 P1=31 : REM ** ACTUALIZA PUERTO 1 DE INTERFACE **
50 P2=19 : REM ** ACTUALIZA PUERTO 2 DE INTERFACE **
60 PP=23 : REM ** ACTUALIZA AMBOS PUERTOS DE LA INTERFACE **
70 LC=1:RB=2:SR=4:LR=8:FZ=16:FC=32:FD=64:CH=128
80 POKE PU,P1:POKE SS,LC+RB+FZ
90 POKE PU,P2:POKE SS,SR+LR+RB
```

READY.

READY.

```
200 REM ***** RUTINA 1 *****
201 REM ***** RUTINA DE LECTURA DEL CONTROL PORT 1 *****
205 POKE 56322,224: REM ** DESHABILITA LECTURA DEL TECLADO **
210 J=PEEK(56321): REM ** LEE CONTROL PORT 1 Y ASIGNA SU VALOR A J **
220 IF (J AND 1)=0 THEN PRINT "BIT 0 ESTA ACTIVADO"
230 IF (J AND 2)=0 THEN PRINT "BIT 1 ESTA ACTIVADO"
240 IF (J AND 4)=0 THEN PRINT "BIT 2 ESTA ACTIVADO"
250 IF (J AND 8)=0 THEN PRINT "BIT 3 ESTA ACTIVADO"
260 IF (J AND 16)=0 THEN PRINT "BIT 4 ESTA ACTIVADO"
270 POKE 56322,255:REM ** REHABILITA LECTURA DEL TECLADO **
280 RETURN
```

READY.

READY.

```
200 REM ***** RUTINA 2 *****
201 REM ***** RUTINA DE LECTURA DE AMBOS PUERTOS *****
205 POKE 56322,224: REM ** DESHABILITA LECTURA DEL TECLADO **
210 J=PEEK(56321):K=PEEK(56320):REM ** VALOR DE CP1 ASIGNADO A J; CP2 A K **
220 IF (J AND 1)=0 THEN PRINT "BIT 0 DEL CP1 ESTA ACTIVADO"
230 IF (J AND 2)=0 THEN PRINT "BIT 1 DEL CP1 ESTA ACTIVADO"
240 IF (J AND 4)=0 THEN PRINT "BIT 2 DEL CP1 ESTA ACTIVADO"
250 IF (J AND 8)=0 THEN PRINT "BIT 3 DEL CP1 ESTA ACTIVADO"
260 IF (J AND 16)=0 THEN PRINT "BIT 4 DEL CP1 ESTA ACTIVADO"
270 IF (K AND 1)=0 THEN PRINT "BIT 0 DEL CP2 ESTA ACTIVADO"
275 IF (K AND 2)=0 THEN PRINT "BIT 1 DEL CP2 ESTA ACTIVADO"
280 IF (K AND 4)=0 THEN PRINT "BIT 2 DEL CP2 ESTA ACTIVADO"
285 IF (K AND 8)=0 THEN PRINT "BIT 3 DEL CP2 ESTA ACTIVADO"
290 IF (K AND 16)=0 THEN PRINT "BIT 4 DEL CP2 ESTA ACTIVADO"
295 POKE 56322,255:REM ** REHABILITA LECTURA DEL TECLADO **
296 RETURN
```

READY.

READY.

```
10 REM *** RUTINA 3 ***
11 REM *** LECTURA DE PUERTOS ***
15 DIM R1$(50),R2$(50),C1$(180),C2$(180)
20 TI$="00000":REM ** INICIALIZA TIEMPO DE LA SESION **
50 FOR I=1 TO 50:REM ** CONTROL # ENSAYOS**
60 FOR K=1 TO 180:REM ** CONTROL # DE BINS **
70 TA=TI:REM ** REGISTRA TIEMPO INICIO DEL BIN **
80 TB=TI:IF TB-TA+19 THEN 80:REM ** CONTROL DURACION DEL BIN**
110 POKE 56322,224
120 J=PEEK(56320):H=PEEK(56321):C1$(K)="X":C2$(K)="X"
130 IF (J AND 1)=0 THEN C1$(K)="A"
140 IF (J AND 2)=0 THEN C1$(K)="B"
150 IF (J AND 4)=0 THEN C1$(K)="C"
160 IF (H AND 1)=0 THEN C2$(K)="A"
170 IF (H AND 2)=0 THEN C2$(K)="B"
180 IF (H AND 4)=0 THEN C2$(K)="C"
181 R1$(I)=RA$(I)+C1$(K):REM** SUMA DE RESP. EN VBLE. ALFANUMERICA **
182 R2$(I)=RB$(I)+C2$(K)
185 POKE 56322,255
190 NEXT K
195 GOSUB 300:REM ** RUTINA DE REFORZAMIENTO **
198 NEXT I
200 GOSUB 500:REM ** RUTINA DE FINALIZACION **
```

READY.

```

500 REM *** RUTINA 5 ***
501 REM ** RUTINA DE REFORZAMIENTO EN P1 POR 4 SEGUNDOS **
510 POKE PU,P1: REM * HABILITA SALIDAS DE PUERTO 1 **
520 TA=TI: REM * REGISTRA TIEMPO DE INICIO DEL PERIODO DE REF. EN TA **
530 POKE SS,SR+LR:REM * ACTIVA DISPENSADOR Y LUZ COMEDERO **
540 TB=TI:REM ** REGISTRA TIEMPO ACTUAL EN TB **
545 REM ** GOSUB * RUTINA LECTURA PUERTO 2 **
550 IF TB<TA+240 THEN 540: REM * COMPARA SI TRANSCURRIERON 4 SEGUNDOS **
560 END

```

READY.

```

1 REM ***** RUTINA 4 *****
2 REM ** LECTURA DE DOS PUERTOS SENALANDO TIEMPO DE EJECUCION **
3 DIM R1%(2400),R1(2400),R2%(2400),R2(2400):REM *VAR %=CODIGO; VAR REAL=TIEMPO
4 REM **** ENTRA RUTINA DE INICIALIZACION DE LA INTERFACE ****
5 REM **** INICIALIZACION DEUN ENSAYO EN CAJA 1: ****
10 POKEPU,P1:POKES,LC+FZ:N1=N1+1
20 TA=TI: REM ** TA ASUME TIEMPO DE INICIO DEL ENSAYO EN CAJA 1 ****
30 POKEPU,P2:POKES,LC+FZ:N2=N2+1
40 TC=TI: REM ** TC ASUME TIEMPO DE INICIO DEL ENSAYO EN CAJA 2 **
50 GOSUB 200
60 END
200 REM ** RUTINA DE LECTURA DE RESPUESTAS : **
205 POKE56322,224
210 J=PEEK(56321):K=PEEK(56320)
215 IF (J AND 1)=0 THEN TB=TA-TI: R1%(N1)=1:R1(N1)=TB
220 IF (J AND 2)=0 THEN TB=TA-TI: R1%(N1)=2:R1(N1)=TB
250 IF (K AND 1)=0 THEN TD=TC-TI: R2%(N2)=3:R2(N2)=TC
260 IF (K AND 2)=0 THEN TD=TC-TI: R2%(N2)=4:R2(N2)=TC
270 RETURN

```

READY.

```

500 REM *** RUTINA 5.1 ***
501 REM ** RUTINA DE REFORZAMIENTOS DISCRETOS **
510 POKE PU,P1: REM * HABILITA SALIDAS DE PUERTO 1 **
520 TA=TI: REM * REGISTRA TIEMPO DE INICIO DEL PERIODO DE REF. EN TA **
530 POKE SS,SR+LR:REM * ACTIVA DISPENSADOR Y LUZ COMEDERO **
535 FOR I=1 TO 5:NEXT I:REM** MANTIENE PULSO **
536 POKE SS,LR:REM * DESACTIVA PULSO Y MANTIENE LUZ DE COMEDERO **
540 TB=TI:REM ** REGISTRA TIEMPO ACTUAL EN TB **
545 REM ** GOSUB * RUTINA LECTURA PUERTO 2 **
550 IF TB<TA+240 THEN 540: REM * COMPARA SI TRANSCURRIERON 4 SEGUNDOS **
560 END

```

READY.

```
300 REM *** RUTINA 6 ***.
301 REM *** GRABADO DE DATOS ***
310 OPEN1,8,1,"FILENAME":REM** ABRE ARCHIVO 'FILENAME' EN UNIDAD DE DISCO **
320 FOR I=1 TO 180
330 PRINT#1,R1$(I):REM * GRABA RESPUESTAS EN CAJA 1 **
340 PRINT#1,R2$(I):REM * GRABA RESPUESTAS EN CAJA 2 **
350 NEXT I
360 END
```

READY.

```
1 REM ***** RUTINA 6.1 *****
2 REM ** GRABACION DE DATOS GENERADOS EN RUTINA 4 **
3 DIM R1%(2400),R1(2400),R2%(2400),R2(2400)
4 REM **** ENTRA RUTINA 4 ****
40 INPUT " NOMBRE DEL ARCHIVO CAJA 1 ";N1$
43 INPUT " NOMBRE DEL ARCHIVO CAJA 2 ";N2$
50 GOSUB 300
60 END
300 REM ** RUTINA PARA GRABAR DATOS **
305 OPEN1,8,1,N1$
315 FOR I=1 TO N1:PRINT#1,R1%(N1):PRINT#1,R1(N1):CLOSE1
320 OPEN1,8,1,N2$
325 FOR I=1 TO N2:PRINT#1,R2%(N2):PRINT#1,R2(N2):CLOSE1
330 RETURN
```

READY.

```

50 REM *** RUTINA 7 ***
51 REM *** RUTINA DE RAZON FIJA ***
60 INPUT"VALOR DE LA RAZON FIJA";FR
200 REM ** RUTINA DE LECTURA DE PUERTOS **
205 POKE 56322,224
210 J=PEEK(56321):K=PEEK(56320)
250 IF (J AND 2)=0 THEN RA=RA+1:REM * INCREMENTO DEL CONTADOR *
251 IF RA=FR THEN GOSUB 500: REM * RUTINA REF. CAJA 1**
260 IF (K AND 1)=0 THEN RB=RB+1:REM * INCREMENTO DEL CONTADOR *
261 IF RB=FR THEN GOSUB 550: REM * RUTINA REF. CAJA 2**
270 GOTO 200

```

READY.

```

48 REM *** RUTINA 7.1 ***
49 REM *** RUTINA DE RAZON VARIABLE ***
50 INPUT"NUMERO DE ENSAYOS A CORRER";NE
51 E1=1 : E2=1:REM * CONTADORES DEL NUMERO DE ENSAYOS **
60 FOR I=1 TO NE
70 PRINT"VALOR DE LA RAZON PARA EL ENSAYO NUMERO ";I:INPUT VR(I)
90 NEXT I
200 REM ** RUTINA DE LECTURA DE PUERTOS **
205 POKE 56322,224
210 J=PEEK(56321):K=PEEK(56320)
250 IF (J AND 2)=0 THEN RA=RA+1:REM * INCREMENTO DEL CONTADOR *
251 IF RA=VR(E1) THEN GOSUB 500: REM * RUTINA REF. CAJA 1**
260 IF (K AND 1)=0 THEN RB=RB+1:REM * INCREMENTO DEL CONTADOR *
261 IF RB=VR(E2) THEN GOSUB 550: REM * RUTINA REF. CAJA 2**
270 GOTO 200

```

READY.

```

48 REM *** RUTINA 7.2 ***
49 REM *** RUTINA DE RAZON VARIABLE/FUNCION RND ***
50 INPUT"VALOR MINIMO DE LA RAZON";LI
60 INPUT"VALOR MAXIMO DE LA RAZON";LS
200 X=(RND(1)*(LS-LI)+LI):RETURN
201 Y=(RND(1)*(LS-LI)+LI):RETURN
202 REM ** RUTINA DE LECTURA DE PUERTOS **
205 POKE 56322,224
210 J=PEEK(56321):K=PEEK(56320)
250 IF (J AND 2)=0 THEN RA=RA+1:REM * INCREMENTO DEL CONTADOR *
251 IF RA=X THEN GOSUB 500: REM * RUTINA REF. CAJA 1**
260 IF (K AND 1)=0 THEN RB=RB+1:REM * INCREMENTO DEL CONTADOR *
261 IF RB=Y THEN GOSUB 550: REM * RUTINA REF. CAJA 2**
270 GOTO 205

```

READY.

```
48 REM *** RUTINA 8 ***
49 REM *** RUTINA DE INTERVALO FIJO ***
50 INPUT"VALOR DEL INTERVALO FIJO";FI
100 TA=TI :REM * TA ASUME EL TIEMPO DE INICIO DEL ENSAYO **
110 TB=TI :REM * TB ASUME EL TIEMPO DE CONTROL DEL INTERVALO **
120 IF TB<TA+FI THEN 110
130 GOSUB 200 :REM ** RUTINA DE LECTURA DE PUERTOS**
200 REM ** GOSUB RUTINA DE LECTURA DE PUERTOS **
```

READY.

```
48 REM *** RUTINA 8.1 ***
49 REM *** RUTINA DE INTERVALO VARIABLE/FUNCION RND ***
50 INPUT"VALOR MINIMO DEL INTERVALO"; LI
60 INPUT"VALOR MAXIMO DEL INTERVALO"; LS
99 TA=TI:TC=TI:REM ** ASUMEN TIEMPO DE INICIO PRIMER ENSAYO **
100 X=INT(RND(1)*(LS-LI)+LI):RETURN
102 Y=INT(RND(1)*(LS-LI)+LI):RETURN
110 TB=TI :REM * TB ASUME EL TIEMPO DE CONTROL DEL INTERVALO **
120 IF TB<TA+X THEN GOSUB 200:REM ** RUTINA LECTURA PUERTO1 **
140 GOSUB 500: REM ** RUTINA REF. CAJA 1 **
160 TD=TI :REM * TC ASUME EL TIEMPO DE CONTROL DEL INTERVALO **
170 IF TD<TC+Y THEN GOSUB 250:REM ** RUTINA LECTURA PUERTO2 **
180 GOSUB 550:REM ** RUTINA REF. CAJA 2
```

READY.

```

699 REM ***** RUTINA 9 *****
700 REM *** CAPTURA DE DATOS ETAPA DE INICIALIZACION ***
701 PRINT: INPUT" ARCHIVO C.1 ";N1$
702 PRINT: INPUT" ARCHIVO C.2 ";N2$
704 PRINT:INPUT" FECHA ";MT$
705 PRINT:INPUT" FASE EXP. ";FE
706 PRINT:INPUT" NUM. ENSAYOS";NE;FOR S=1TO 4:PRINT:NEXTS
708 PRINT" DATOS CORRECTOS ":INPUT" ( S / N )";DC$
710 IF DC="N"THEN700
716 INPUT" LISTO ";CH$
718 PRINT"▽":FORV=1TO12:PRINT:NEXTV:PRINT" SESION EXPERIMENTAL"
720 PRINT:PRINT" EN CURSO ...."
730 RETURN : REM *** REGRESA A PROGRAMA PRINCIPAL ***

```

READY.

```

888 REM **** RUTINA 9.1 *****
889 REM **** RUTINA DE INTERACCION AL FIN DE LA SESION / CAJA 1 *****
890 R1%(1)=12:R1(1)=N1:REM *** CODIGO QUE INDICA TOTAL DE EVENTOS CAJA 1
891 PRINT"▽":PRINT:PRINT:PRINT" TOTAL DE EVENTOS SUJETO:";N1$;" = ";N1:PRINT
892 PRINT " TOTAL DE RESPUESTAS ";N1:PRINT
893 PRINT" DESEAS VER LOS DATOS DE ";N1$;
895 INPUTDU$:IF DU$<>"S"THEN 913 TOTAL DE EVENTOS SUJETO:";N1$;" = ";N1:PRINT
896 PRINT"NO.EV. ","EVENTO","TIEMPO"
897 PRINT:PRINT: FOR X=1 TO N1
899 PRINTX,R1%(X),R1(X):NEXTX:PRINT:PRINT:PRINT:INPUT" NUEVAMENTE ";HI$
900 IF HI$="S"GOTO 891
913 PRINT:INPUT" DESEAS GRABARLOS ";JC$
915 IF JC$<>"S" THEN 922
916 PRINT"▽":FORS=1TO5:PRINT:NEXTS:PRINT" ( INSERTA EL DISCO )"
917 FORS=1TO5:PRINT:NEXTS:INPUT" LISTO ";WW$:FOR S=1TO7:PRINT:NEXTS
918 PRINT" POR FAVOR ESPERA ...."
919 OPEN1,8,1,N1$:FOR X=1 TO R
920 PRINT#1, R1%(X):PRINT#1, R1(X)
921 NEXTX: CLOSE1
922 PRINT"▽":PRINT:PRINT:INPUT" DESEAS IMPRIMIRLOS ";DO$
923 IF DO$<>"S" THEN 950
925 OPEN1,4:PRINT#1,R1$,MT$, "FASE ";FE
928 PRINT#1:PRINT#1,"NO.EV.", "EVENTO", "TIEMPO":PRINT#1
929 FOR X=1 TO N1:PRINT#1,X,R1%(X),R1(X)
931 NEXTX:CLOSE1
932 RETURN : REM *** REGRESA A PROGRAMA PRINCIPAL *****

```

READY.

```
1 REM ***** RUTINA 11 : ENTRENAMIENTO EN COMEDERO PARA CAJA 2 *****
2 POKE56578,63:POKE56579,255:POKE56323,0:LC=1:CD=16:CI=32
3 PU=56576:SS=56577:P1=31:P2=19:P3=23
4 PRINT"°SC°":INPUT"NUMERO DE ENSAYOS :";NE
5 PRINT"°SC°":INPUT"DURACION DEL PERIODO DE REF. :";TR
10 TI$="000000":R=0:X=1
11 ON X GOSUB 15,16,26,30
12 GOTO11
14 POKEPU,P2:POKESS,0:END
15 POKEPU,P2:POKESS,LC:X=2:RETURN:REM **PRENDE LUZ GENERAL **
16 REM ** LOS COMEDEROS SON ACTIVADOS DESDE EL TECLADO ***
17 PRINT"PRESIONA ( 1 ) PARA ACTIVAR COM.DERECHO ":PRINT:PRINT
18 PRINT"PRESIONA ( 2 ) PARA ACTIVAR COM.IZQUIER ":PRINT:PRINT
20 LE=PEEK(56321)
21 IF LE<>254 THEN 23
22 X=3:T1=TI:POKEPU,P2:POKESS,CD
23 IF LE<>247 THEN 25
24 X=3:T1=TI:POKEPU,P2:POKESS,CI
25 RETURN
26 F1=TI:IF F1<T1+TR*60 THEN28: REM *** EL COM. SE ACTIVA DURANTE 4 SEGS.
27 X=4:R=R+1
28 RETURN
30 IF R>=NE+1:THEN 14 : REM *** EL CICLO SE REPITE NE VECES ***
32 X=1:RETURN
```

READY.

# ALGORITMO GENERAL

## 1) RUTINA DE INICIALIZACION

- Declarar variables requeridas para habilitar la interface.
- Declarar variables requeridas por el procedimiento experimental.
- Subrutina de solicitud de valores para variables a definir al inicio de la sesión.
- Verificación de datos.

## 2) INICIALIZACION DE LA SESION

- Inicialización del reloj.
- subrutina de control de tiempos.

## 3) CONTROL DE ETAPAS DE LA SESION

- Subrutinas de control de ensayos y bins.
- Presentación de estímulos de acuerdo con la etapa de la sesión.
- subrutina de lectura de puertos.
- chechar condicionalidades de respuestas y/o tiempo.
- si procede, acceso a subrutina de reforzamiento.

## 4) SUBROUTINA DE REFORZAMIENTO

- Inicio de tiempo de reforzamiento.
- Presentación de estímulos operativos.
- Si procede, reinicializar variables e incrementar contadores.
- Registrar tiempo y comparar.
- Si procede, efectuar lectura de puertos.
- Tiempo de reforzamiento cumplido, volver a etapa 3.

# ALGORITMO GENERAL

## 5) RUTINA DE FINALIZACION

- Desactivación de líneas de la interface.
- Rehabilitación de teclado.
- Presentación de opciones:
  - Grabar archivos.
  - Imprimir datos generados.
  - Presentar datos en pantalla.
  - Efectuar algun análisis.
- END

# PROCEDIMIENTO

## ENTRENAMIENTO EN COMEDERO

1) Definir cuántas veces voy a entrenar en comedero durante esta sesión.

2) Definir el período de acceso al reforzamiento.

3) Inicializar la sesión.

4) Lectura permanente de los puertos de entrada.

5) Detectar si hay respuestas:

- Si no hay, entonces volver a leer puertos.
- Si hay respuestas, entonces comparar.
  - Si es 1, entonces activar rutina ref. 1.
  - Si es 2, entonces activar rutina ref. 2.

6) Rutina de reforzamiento 1.

- Inicializar tiempo de ref. rutina 1.
- Activar foco comedero y comedero 1.
- Registrar tiempo transcurrido.
- Comparar tiempo transcurrido con período de reforzamiento.
  - Si no ha pasado el período programado, entonces volver a registrar el tiempo transcurrido.
  - Si ya se cumplió el periodo, entonces:

## PROCEDIMIENTO

- Descontar un ciclo.
- Si ya se cumplieron los ciclos programados, entonces finalizar la sesion.
- Si no se han cumplido los ciclos, entonces volver a punto 4.

## PROGRAMA PARA CORRER EL EXPERIMENTO

```

1 REM ***** RATAS/NIETO" *****
2 POKE 53280,1:POKE 53281,1:POKE56578,63:POKE56579,255:POKE56323,0
3 PU=56576:SS=56577:P1=31:P2=19:PP=23
4 GOSUB 600
14 DIM RA$(50),RB$(50),C1$(180),C2$(180)
15 HL=16:LC=2:CH=1:PE=4
16 IF OP%=1 THEN SR=17:DR%=0:LC=16
20 IF OP%=2 THEN SR=6:LC=2:DR%=240
40 TI$="000000"
45 POKE PU,PP:POKESS,HL
50 FOR I=1 TO 50
60 FOR K=1 TO 180
70 TA=TI
80 TB=TI:IF TB<TA+19 THEN 80
110 POKE 56322,224
120 J=PEEK(56320):H=PEEK(56321):C1$(K)="X":C2$(K)="X"
130 IF (J AND 1)=0 THEN C1$(K)="A"
140 IF (J AND 2)=0 THEN C1$(K)="B"
150 IF (J AND 4)=0 THEN C1$(K)="C"
160 IF (H AND 1)=0 THEN C2$(K)="A"
170 IF (H AND 2)=0 THEN C2$(K)="B"
180 IF (H AND 4)=0 THEN C2$(K)="C"
181 RA$(I)=RA$(I)+C1$(K)
182 RB$(I)=RB$(I)+C2$(K)
185 POKE 56322,255
190 NEXT K
195 GOSUB 300
198 NEXT I
200 GOSUB 400
300 POKE PU,PP:POKESS,SR
310 TA=TI: FOR P=1 TO IM:NEXTP
320 POKE PU,PP:POKESS,LC
330 TB=TI:IF TB<TA+DR% THEN 330
340 POKE PU,PP:POKESS,HL
360 RETURN
400 PRINT"°SC'":INPUT"QUIERES GRABAR LOS DATOS";Q$
401 IF Q$="N" THEN 500
405 PRINT"°SC'":PRINT:PRINT"PONER DISCO DE DATOS"
410 PRINT:PRINT:INPUT" L I S T O (S/N)";W$
415 IF W$="N" THEN 400
416 PRINT"°SC'": PRINT:PRINT"G R A B A N D O"
420 A1$=S1$+DA$: A2$=S2$+DA$
430 OPEN1,8,1,A1$
440 FOR I=1 TO 50: PRINT#1,RA$(I):NEXTI
450 CLOSE1
460 OPEN1,8,1,A2$
465 FOR I=1 TO 50: PRINT#1,RB$(I):NEXTI
470 CLOSE1
480 PRINT"°SC'":PRINT:INPUT"QUIERES VOLVER A GRABAR ";G$
481 IF G$="S" THEN 400
500 POKE PU,PP:POKESS,0:END

```

READY.

continúa...

```

500 POKE PU,PP:POKESS,0:END
600 PRINT"°SC³":PRINT:INPUT"      SUJETO CAJA 1 :";S1$
610 PRINT:PRINT:INPUT"      SUJETO CAJA 2 :";S2$
620 PRINT:PRINT:PRINT"      GRUPO EXPERIMENTAL : "
630 PRINT:PRINT SPC(10):PRINT"CHOQUE ( 1 )"
640 PRINT:PRINT SPC(10):PRINT"PELLETS ( 2 )"
645 PRINT:PRINT SPC(15):INPUT"OPCION ( 1/2 )";OP%
646 PRINT:PRINT:PRINT:INPUT"      DURACION DEL PULSO :";IM
650 PRINT:PRINT:PRINT:INPUT"      FECHA (AAMDD) :";DA$
660 PRINT:PRINT SPC(12):INPUT" DATOS CORRECTOS (S/N)";B$
661 IF B$="N" THEN 600
670 PRINT"°SC³":PRINT:PRINT"*** PRENDER FUENTES +28, +5, CHOQUES"
675 PRINT:PRINT:PRINT"*** METER ANIMALES"
680 PRINT:PRINT:PRINT"*** PONER DISCO DE DATOS"
685 PRINT:PRINT:PRINT"*** CONECTAR CONTROL PORT 1"
690 PRINT:PRINT:PRINT:PRINT:PRINT SPC(12):INPUT"L I S T O (S/N)";V$
691 IF V$="N" THEN 600
695 PRINT"°SC³":FOR Y=1TO12:NEXTY:PRINT SPC(12): PRINT" SESION EXPERIMENTAL"
699 PRINT:PRINT:PRINT"      EN CURSO"
700 RETURN

```

READY.

```

300 REM ** ANALISIS DE DATOS **.          PROGRAMA DE ANALISIS DE DATOS
301 CLR
305 DIM AB%(60,10), AE%(60), AS%(10)
306 DIM BB%(60,10), BE%(60), BS%(10)
307 DIM CB%(60,10), CE%(60), CS%(10)
308 DIM XB%(60,10), XE%(60), XS%(10)
350 PRINT"SC":PRINT:INPUT" NOMBRE DEL ARCHIVO ";NA$:OPEN1,8,0,NA$
362 PRINT:PRINT"SUJETO : ";LEFT$(NA$,3)
363 PRINT:PRINT"FECHA  : ";RIGHT$(NA$,6)
366 FOR I=1TO60
367 FOR L=1TO10
368 FOR K=1TO18
369 GET#1,A$
370 IF A$="A" THEN AB%(I,L)=AB%(I,L)+1:AE%(I)=AE%(I)+1:AS%(L)=AS%(L)+1
371 IF A$="B" THEN BB%(I,L)=BB%(I,L)+1:BE%(I)=BE%(I)+1:BS%(L)=BS%(L)+1
372 IF A$="C" THEN CB%(I,L)=CB%(I,L)+1:CE%(I)=CE%(I)+1:CS%(L)=CS%(L)+1
373 IF A$="X" THEN XB%(I,L)=XB%(I,L)+1:XE%(I)=XE%(I)+1:XS%(L)=XS%(L)+1
375 NEXTK:NEXTL:NEXTI
376 CLOSE1
377 PRINT"SC":PRINT"TOTAL DE RESPUESTAS POR BINS":PRINT:
378 PRINT" BIN1 BIN2 BIN3 BIN4 BIN5 BIN6 BIN7 BIN8 BIN9 BIN10 TOTAL"
379 PRINT"PALANCA : "
380 FOR L=1 TO 10:Z=AS%(L):GOSUB900
382 PRINT Z;
383 AT%=AT%+AS%(L)
385 NEXTL
386 PRINTAT%:PRINT:
389 PRINT"PUERTA  : "
390 FOR L=1 TO 10:Z=BS%(L):GOSUB 900
392 PRINTZ;
393 BT%=BT%+BS%(L)
395 NEXTL
396 PRINTBT%:PRINT:
400 PRINT"BEBEDERO: "
410 FOR L=1 TO 10:Z=CS%(L): GOSUB900
422 PRINTZ;
433 CT%=CT%+CS%(L)
435 NEXTL
436 PRINTCT%:PRINT:
440 PRINT"NO RESP. : "
450 FOR L=1 TO 10:Z=XS%(L):GOSUB900
462 PRINTZ;
463 XT%=XT%+XS%(L)
466 NEXTL
467 PRINTXT%:PRINT:
470 RT%=AT%+BT%+CT%+XT%
471 INPUT"QUIERES VER TOTAL DE RESPUESTAS";P$
472 IF P$="N" THEN 700
480 PRINT:PRINT"SUJETO : ";LEFT$(NA$,3)
490 PRINT:PRINT"FECHA  : ";RIGHT$(NA$,6):PRINT:
500 PRINT"      TOTAL DE RESPUESTAS POR SESION":PRINT:
509 PRINT"          TOTAL      TIEMPO      FRECUENCIA"
510 PRINT"          RESPUESTAS  SEG.        RELATIVA ":PRINT:

```

READY.

continua....

PROGR. ANALISIS DE DATOS

```

520 PRINT"PALANCA :";
521 Z=AT%:GOSUB910:PRINTZ;
522 Z=INT(AT%/3):GOSUB910:PRINTZ;
523 Z=INT(100*AT%/RT%):GOSUB910:PRINT SPC(3):PRINTZ:PRINT:
530 PRINT"PUERTA :";
531 Z=BT%:GOSUB910:PRINTZ;
532 Z=INT(BT%/3):GOSUB910:PRINTZ;
533 Z=INT(100*BT%/RT%):GOSUB910:PRINT SPC(3):PRINTZ:PRINT:
540 PRINT"BEBEDERO :";
541 Z=CT%:GOSUB910:PRINTZ;
542 Z=INT(CT%/3):GOSUB910:PRINTZ;
543 Z=INT(100*CT%/RT%):GOSUB910:PRINT SPC(3):PRINTZ:PRINT:
550 PRINT"NO RESP.:";
551 Z=XT%:GOSUB910:PRINTZ;
552 Z=INT(XT%/3):GOSUB910:PRINTZ;
553 Z=INT(100*XT%/RT%):GOSUB910:PRINT SPC(3):PRINTZ:PRINT:
600 INPUT"QUIERES VER NUEVAMENTE LOS DATOS";Q$
610 IF Q$="S" THEN 377
700 PRINT:INPUT"QUIERES IMPRIMIR LOS DATOS ";PA$
710 IF PA$="N" THEN 800
712 OPEN1,4
713 PRINT#1:PRINT#1,"SUJETO : ";LEFT$(NA$,3)
714 PRINT#1:PRINT#1,"FECHA : ";RIGHT$(NA$,6):PRINT#1
715 PRINT#1:PRINT#1,"TOTAL DE RESPUESTAS POR BINS":PRINT#1
716 PRINT#1,"BIN1 BIN2 BIN3 BIN4 BIN5 BIN6 BIN7 BINS BIN9 BIN10 TOTAL"
719 PRINT#1,"PALANCA :";
720 FOR L=1 TO 10:Z=AS%(L):GOSUB920
722 PRINT#1,Z;
725 NEXTL
726 PRINT#1,AT%:PRINT#1
729 PRINT#1,"PUERTA :";
730 FOR L=1 TO 10:Z=BS%(L):GOSUB 920
731 PRINT#1,Z;
734 NEXTL
736 PRINT#1,BT%:PRINT#1
737 PRINT#1,"BEBEDERO:";
739 FOR L=1 TO 10:Z=CS%(L):GOSUB920
740 PRINT#1,Z;
742 NEXTL
743 PRINT#1,CT%:PRINT#1
745 PRINT#1,"NO RESP.:";
747 FOR L=1 TO 10:Z=XS%(L):GOSUB 920
749 PRINT#1,Z;
753 NEXTL
755 PRINT#1,XT%:PRINT#1
756 PRINT#1:PRINT#1:PRINT#1
760 PRINT#1," TOTAL DE RESPUESTAS POR SESION":PRINT#1
761 PRINT#1," TOTAL TIEMPO FRECUENCIA"
762 PRINT#1," RESPUESTAS SEG. RELATIVA ":PRINT#1
764 PRINT#1,"PALANCA :";
766 Z=AT%:GOSUB930:PRINT#1,Z;
768 Z=INT(AT%/3):GOSUB930:PRINT#1,Z;
769 Z=INT(100*AT%/RT%):GOSUB930:PRINT#1, SPC(3):PRINT#1,Z:PRINT#1
770 PRINT#1,"PUERTA :";

```

continúa...

READY.

```
771 Z=BT%:GOSUB930:PRINT#1,Z;
772 Z=INT(BT%/3):GOSUB930:PRINT#1,Z;
773 Z=INT(100*BT%/RT%):GOSUB930:PRINT#1, SPC(3):PRINT#1,Z:PRINT#1
774 PRINT#1,"BEBEDERO:";
775 Z=CT%:GOSUB930:PRINT#1,Z;
777 Z=INT(CT%/3):GOSUB930:PRINT#1,Z;
778 Z=INT(100*CT%/RT%):GOSUB930:PRINT#1, SPC(3):PRINT#1,Z:PRINT#1
779 PRINT#1,"NO RESP.:";
780 Z=XT%:GOSUB930:PRINT#1,Z;
781 Z=INT(XT%/3):GOSUB930:PRINT#1,Z;
782 Z=INT(100*XT%/RT%):GOSUB930:PRINT#1, SPC(3):PRINT#1,Z:PRINT:
785 CLOSE1
786 GOTO 600
800 END
900 IF Z#10 THEN PRINT SPC(3)
901 IF Z#9 AND Z#100 THEN PRINT SPC(2)
902 IF Z#99 AND Z#1000 THEN PRINT SPC(1)
903 IF Z#999 THEN PRINT SPC(0)
904 RETURN
910 IF Z#10 THEN PRINT SPC(5)
911 IF Z#9 AND Z#100 THEN PRINT SPC(4)
912 IF Z#99 AND Z#1000 THEN PRINT SPC(3)
913 IF Z#999 THEN PRINT SPC(2)
914 IF Z#9999 THEN PRINT SPC(1)
915 RETURN
920 IF Z#10 THEN PRINT#1,SPC(3)
921 IF Z#9 AND Z#100 THEN PRINT#1,SPC(2)
922 IF Z#99 AND Z#1000 THEN PRINT#1,SPC(1)
923 IF Z#999 THEN PRINT#1,SPC(0)
924 RETURN
930 IF Z#10 THEN PRINT#1,SPC(5)
931 IF Z#9 AND Z#100 THEN PRINT#1,SPC(4)
932 IF Z#99 AND Z#1000 THEN PRINT#1,SPC(3)
933 IF Z#999 THEN PRINT#1,SPC(2)
934 IF Z#9999 THEN PRINT#1,SPC(1)
935 RETURN
```

READY.

## BIBLIOGRAFIA

Angerhausen, M., Becker, A., Englisch, L., y Gerits, K. (1984) The Anatomy of the Commodore 64. Michigan: Abacus Software Inc.

Butler, F.E. (1980) MicroSKED. Behavior Research Methods and Instrumentation, 12, (2), 152-154.

Commodore Business Machines, Inc. (1983) Commodore 64 Programmer's Reference Guide. Wayne, PA: Commodore Business Machines Inc. and Howard W. Sams & Co., Inc.

Davies, H. y Hubbard J. (1972) An analysis of superstitious behavior in the rat. Behaviour 42: 1-12.

Falk, J.L. (1971) The nature and determinants of adjunctive behavior. Physiol. Behav. 6: 577-588.

Ferster, C.B. y Skinner, B.F. (1957) Schedules of Reinforcement. New York: Appleton Century Crofts.

Heilborn, J., y Talbott, R. (1985) Commodore 64, Guia del usuario. Madrid: OSBORNE/Mc Graw-Hill.

Leemon, S. (1984) Mapping the Commodore 64. Greensboro, NC: COMPUTE! Publications Inc.

Greist, J.H. y Klein, M.H. (1980) Computer programs for patients, clinicians and researchs in psychiatry. En J.B Sidowsky, J.H. Johnson y T.A. Williams (Eds.) Technology in mental health care delivery system. Norwood, N.J.: Ablex.

Lang, P.J. (1980) Behavioral treatment and bio-behavioral assesment. En J.B. Sidowsky y col. (op. cit.).

Perera, T.B. (1981) A simple BASIC language laboratory control system for microcomputers patterned after SKED. Behavior Research Methods & Instrumentation, 13, (2), 281-282.

Rayfield, F. (1982) Experimental control and data acquisition with BASIC in the Apple computer. Behavior Research Methods & Instrumentation, 14, (4), 409-411.

Ribes, E. y Chávez R. (1988) Efectos de la comida no contingente en la conducta libre de la rata: consideraciones

sobre el analisis del segmento de respuesta. Revista Mexicana de Analisis de la Conducta. (en prensa).

Sanger, D.J. y Blackman D.E. (1976) Rate-dependent effects of drugs: A review of the literature. Pharmac. Biochem. Behav. 4: 73-83.

Selmi, P.M., Klein, M.H. Greist, J.H., y Harris, W.G. (1982) An investigation on computer-assited cognitive-behavior therapy in the treatment of depression/ Behavior Research Methods & Instrumentation, 14, (2), 181-185.

Snapper, A.G. (1974) Alternative versions of SKED: Current systems and future plans. Behavior research Methods & Instrumentation, &, 171-173.

Skinner, B.F. (1966) Operant Behavior. En Honig, W.K. (Ed.) OPERANT BEHAVIOR: Areas of research and application. New York: Meredith Corporation.

Skinner, B.F. (1974) About Behaviorism. New York: Knopf.

Staddon, J.E.R. (1977) Schedule-induced behavior. En Honig, W.K. y Staddon, J.E.R. (Eds.) Handbook of Operant Behavior. New York: Prentice- Hall.

Staddon, J.E.R. y Ayres, S.L. (1975) Sequential and temporal properties of behavior induced by a schedule of periodic food delivery. Behaviour. 54: 26-49.