

# Control Remoto de Formato Comercial

C. Aquilino Cervantes Avila  
Alumno de la Maestría del CINTEC-IPN.

**E**l propósito inicial de este trabajo fue el de manejar un sistema mínimo 80188 por medio de un control remoto, como opción adicional al teclado. De esta forma, se parte de las experiencias de los alumnos Marcos Germán Aureliano Guadalupe y Agustín Cruz Contreras, quienes en la materia de prerequisite "Introducción a los Lenguajes de Alto Nivel", impartida por el profesor Miguel Angel Partida Tapia (M. en C.) para la Maestría de este Centro, investigaron el formato que presenta la codificación de un control remoto comercial.

Realmente lo que se diseña en este desarrollo es un teclado común del tipo matriz, pero se aumenta un registro para captura de la tecla, un registro de corrimiento y un codificador que hace el cambio de formato serie al formato comercial. El control remoto se desarrolla sobre un dispositivo MAPL 128; esto da la ventaja de trabajar en un espacio pequeño para la tarjeta del control remoto, y asegura un tiempo de vida para la batería de alimentación mayor a que si fuera desarrollada en base a lógica normal.

## Descripción general

El teclado trabaja de la siguiente manera: existe un contador binario

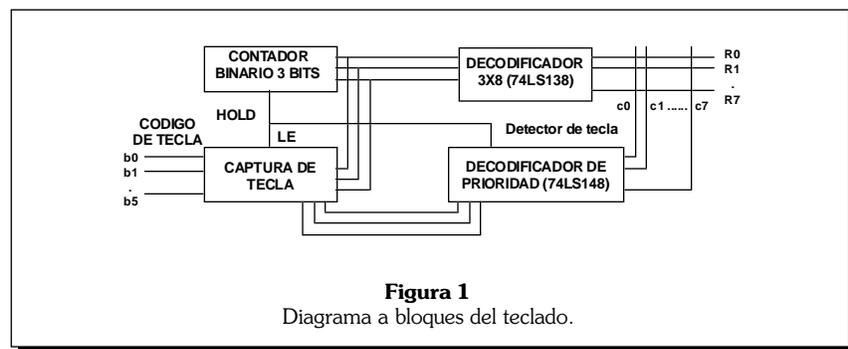
de tres bits con captura (**hold**), que alimenta un decodificador (74LS 138) de 3 X 8, activo bajo. Así, se logra el efecto de "búsqueda de tecla" en las salidas del decodificador; por otra parte, se tiene un decodificador de prioridad (74LS 148) 8 X 3, el cual regresa un número binario correspondiente a la entrada que recibe un nivel bajo. Además, una señal indica si alguna línea de entrada recibió un estado bajo.

Con estos tres dispositivos se forma un teclado colocando las líneas de entrada del codificador de prioridad como columnas y las salidas del decodificador como renglones, agregando un registro de captura de tecla.

su vez al registro de captura, asegurando que se tiene la información tanto de columnas como de renglones.

Para evitar errores debido al ruido eléctrico producido por los rebotes de las teclas, se agrega una red RC en la línea de detección de tecla, eliminando este problema. Con ello, la línea de detección se configura como salida y también como entrada al MAPL128.

La tecla de detección también se usa para indicar un inicio de sincronía, para que el receptor conozca el momento en que está llegando información y no ruido.



NOTA: si existe algún problema en el reconocimiento de tecla, se recomienda que las columnas posean resistencias de levantamiento

Así, se realiza el arreglo para que la línea que detecta algún bajo también detenga al contador y dispare a

La información contenida en el registro de captura, es transportada por un registro de corrimiento para obtener un formato serie. Cada bit es manipulado por el codificador formato comercial y enviado por un emisor óptico.

El codificador comercial transporta la información SOLO EN LA PARTE BAJA de la señal, esto es, en la parte baja de los pulsos tenemos que:

	Representa un inicio o sincronía
	Representa un estado bajo
	Representa un estado alto

Es posible observar que el nivel alto representa un espacio, el estado bajo representa dos espacios y un inicio de transmisión es lo mismo a tres espacios. Aunado a esto es necesario indicar que la codificación es del tipo retorno a uno y, además, el tiempo de permanencia en uno es constante y proporcional a un espacio. El número de pulsos que transmite un control remoto comercial es de 12 pulsos por tecla y en el momento en que no se está enviando información, la salida se encuentra en nivel alto.

Debido a que se tienen 8 columnas y 8 renglones, solo se tienen 6 bits de información y el paquete de datos que recibe el receptor necesita tener 12. La solución a este problema fue recorrer nuevamente los 6 bits de información.

Para hacer más flexible al control remoto, se presenta la opción de una tecla que permita doblar la capacidad del teclado (SHIFT); para este fin se emplea un flip-flop tipo 'D' configurado para que cambie de nivel cada vez que se oprima una tecla externa; este bit se pasa por el registro de corrimiento y se entrega al codificador formato comercial. Entonces, lo que hace el sistema realmente es recorrer primero los 6 bits de información de tecla, recorrer nuevamente los primeros 5 bits y agregar al final, el bit que corresponde a SHIFT.

Una vez que se detecta una tecla, aunque se oprima otra, no se hace caso de ella hasta el término del envío de la información. Si se oprimen dos teclas a la vez solo será reconocida la de mayor prioridad.

Al término de envío de información, se restablece nuevamente la cuenta del contador para seguir con la "búsqueda de teclas".

Si se mantiene una tecla oprimida la información seguirá siendo enviada al receptor; esto permite que sea posible observar la señal en un osciloscopio y crear la función conocida en teclados como "type-matic".

### **Descripción de la programación del MAPL**

Para programar el MAPL 128, se utilizó el método de tablas de verdad con paginación manual, usando el paquete OPAL propiedad de NATIONAL SEMICONDUCTORS. Es necesario recalcar que el OPAL no soporta más de un tipo de tabla de verdad, por lo que, si se desea verificar el ejemplo, es necesario utilizar métodos alternativos o compilar en forma parcial las diferentes tablas, y emplear los módulos .EQN.

En la página 000 del MAPL, se localiza un contador binario de 3 bits unido en forma directa a un decodificador 3x8. Se puede observar que cuando la señal "tec" es llevada a cero, se hace una captura de la cuenta que posee el contador, y se hace un cambio a la página 001.

La señal "tec" indica si alguna tecla fue activada; para esto, en el diagrama del control remoto se une la señal de salida "detec" con la señal de entrada "tec". La señal "detec" envía un estado bajo si al menos una de las entradas c0 - c7 es

llevada a cero. Detec es activa en las 8 páginas.

En la página 001 se localiza un decodificador de prioridad el cual permite verificar cual de las señales c0 - c7 fue llevada a un estado bajo; si fueron varias las teclas activadas solo se tomará la de mayor prioridad. El registro de captura que se menciona en el diagrama de bloques se implementa dentro de los propios módulos descritos, gracias a que tanto las salidas como los nodos internos se encuentran registrados.

El registro de corrimiento obliga a tener en una sola línea la información de todos los bits del registro de captura en un instante de tiempo diferente; para lograr este objetivo, se deja que el registro de captura se encuentre activo en la página 010 y se maneja el programa de tal manera que se toma el primer bit del registro de captura y se pasa al codificador comercial; una vez que se terminó de codificar, se regresa nuevamente al registro de captura, para tomar el siguiente bit, se codifica y se regresa nuevamente al registro de captura, y así sucesivamente. Como la información de la tecla activada se encuentra en 6 bits y el formato comercial pide 12, entonces se recorren por segunda vez los mismos 6 bits.

Así, se utilizan las variables de control a4,a3,a2,a1 y a0 en forma conjunta con los registros de paginación p0, p1 y p2, para elegir la combinación y página adecuada para seguir el procedimiento descrito.

Para marcar el inicio del dato, la tabla que codifica éste posee dos puntos de entrada: el inicial, en donde marca el inicio de dato la combinación a4,a3,a2,a1 y a0 = 00 000 y para el segundo recorrido, la combinación a4,a3,a2,a1 y a0 = 01 000; para insertar la información

del bit de duplicación de teclas "shift", se intercambia esta información, con el ultimo bit por transmitir del segundo recorrido. Con esto, la información que lee el receptor es: 6 bits de información, 5 bits de redundancia y 1 bit de duplicación de teclas.

Por ultimo se indica que en las tablas de codificación comerciales, se emplean elementos de más para asegurar que los tiempos de intercambio y selección de página no afecten la señal de salida al receptor.

Una vez echo esto, falta escoger el emisor infrarrojo a utilizar, así como el detector óptico para la etapa de recepción; por tanto, del manual de optoelectrónica citado en la bibliografía se eligen los dispositivos, MLED91 para el emisor y MRD911 para el receptor.

El MLED91 es un led infrarrojo, dentro del rango de  $\lambda = 940$  nm. y posee además las características de ángulo de emisión de  $60^\circ$  y potencia de salida de 2.5 mW a 50 mA de corriente. Por su parte, el MRD911, es un fototransistor darlington NPN, que también se encuentra en el rango  $\lambda = 940$  nm y posee una corriente en iluminación, de 25mA para  $V_{CE} = 5V$  y  $H = 0.5$  mW/cm<sup>2</sup>.

El diodo emisor infrarrojo NO se puede conectar en forma directa al MAPL128, esto es debido a que el dispositivo mencionado sólo puede drenar corriente del orden de microamperes; por lo tanto, es necesario colocar un transistor como drenador de corriente.

Para la etapa de recepción se tiene que, por si mismo, el fototransistor no proporciona la señal suficiente para el manejo correcto del circuito. La solución fue emplear un amplificador operacional LM324 para obtener la amplificación nece-

saria para el funcionamiento adecuado.

Para el circuito mostrado se trabajó en forma satisfactoria hasta una distancia de aproximadamente 10 m. Existen amplificadores dedicados para el uso de fotosensores infrarrojos; el problema es que para el funcionamiento adecuado de estos últimos, es necesaria una fuente de 12V. Las ventajas que presentan los amplificadores dedicados son muchas, pero el sistema desarrollado esta atado a un sistema mínimo 80188 que por diseño posee solamente una fuente de 5V; para evitar el utilizar dos fuentes diferentes, o algún otro dispositivo para lograr los 12 V, se optó por el amplificador operacional descrito, LM324, que opera con una sola fuente de 5V.

---

### Conclusiones

---

Para este trabajo se utilizo un formato SONY, solo para ser consistentes con un control remoto comercial, sin embargo, no se tienen completamente las características de codificación de SONY, por lo que el control remoto NO ES compatible con SONY.

Para fines prácticos es más fácil el codificar solo los 6 bits de información que proporciona el teclado y funciona igualmente bien que con el formato aquí seguido. Si se desea mayor seguridad y además no se necesita de una señal de duplicación de teclas, seria conveniente el codificar 2 veces la misma información para averiguar si existió algún error en la transmisión.

Además, sin romper el esquema general, es posible el codificar un numero menor de teclas para aumentar líneas de control parecidas a "SHIFT" y así trabajar en base a las necesidades propias.

La elección del dispositivo MAPL 128 para el desarrollo de esta practica, brinda un espacio pequeño en tarjeta, además de consumir poca corriente y alargar con esto la vida de la batería empleada. El control remoto fue alimentado con una batería standard de 9 voltios y el teclado utilizado es una adaptación de una calculadora.

El ajuste de la señal de reloj se hizo en base a un dispositivo NE555 configurado como multivibrador asintótico con frecuencia variable.

Para terminar, se mencionará que las pruebas efectuadas con los dispositivos indicados en el diagrama brindaron un trabajo satisfactorio hasta para separaciones de 10m, si se apunta en forma directa al emisor del control remoto con el receptor.

# Control Remoto de Formato Comercial

```

PROGRAMA DE APLICACION

BEGIN HEADER

                                IPN

                                CENTRO DE INVESTIGACION TECNOLOGICA EN COMPUTACION

                                EDIFICIO DE GRADUADOS DE UPIICSA

                                CALLE TE NO. 950 COL. GRANJAS MEXICO, MEXICO D.F.
                                    CP 08400

                                ALUMNO: AQUILINO CERVANTES AVILA

DESCRIPCION:
Se implementa un control remoto formato SONY, en un MAPL128

END HEADER

BEGIN DEFINITION
Device MAPL128;

Inputs  clk=1,tec,c0,c1,c2,c3,c4,c5,c6,c7,sht;
        {c0 - c7 son las columnas, sht permite duplicar las }
        {teclas, y tec es una retroalimentación externa de detec}

Feedbacks (jk,hold) b1,b2,b3,b4,b5,bs; {bits de captura de información b0 - b5 }
Feedbacks (jk,hold,set) out;           { salida del control remoto }
Outputs (reg) r0,r1,r2,r3,r4,r5,detec;  { se declaran 6 renglones }
Feedbacks (buried) p0=37,p1=38,p2=39; { registros de paginación }
Feedback (jk,hold,buried) a0,a1,a2,a3,a4,bn,b0;
        { bits de control bs y bn son bits de corrimiento }

END DEFINITIONS

BEGIN TRUTH_TABLE {Esta tabla tiene el control de la tecla SHIFT }
TTIN sht,bs,p2,p1,p0; {la tecla shift maneja un registro de manera que}
TTOUT bs,p2,p1,p0;    {a cada cambio de "sht" a cero, el bit "bs" cambie}
                    { a su complemento. si "sht" se encuentra en uno }
1 - 000 ? 000       {el valor de "bs" es capturado}
01 000 0 000
00 000 1 000
END TRUTH_TABLE

BEGIN EQUATIONS { Esta ecuación detecta si se oprimió una tecla }
detec = c0 * c1 * c2 * c3 * c4 * c5 * c6 * c7;
        {detec se hace cero si cualquier c0..7 se vuelve cero }

END EQUATIONS

BEGIN TRUTH_TABLE { Esta tabla realiza a un contador binario de 3 bits atado }
                    { a un decodificador 3x8 }
TTIN tec,b2,b1,b0,p2,p1,p0;
TTOUT b2,b1,b0,r0,r1,r2,r3,r4,r5,p2,p1,p0;
0 --- 000 ??? ----- 001 {si "tec" es llevado a cero, entonces se detiene}

```

```

1 111 000 000 011111 000 {la cuenta y se pasa a la página siguiente}
1 000 000 001 101111 000
1 001 000 010 110111 000
1 010 000 011 111011 000 { ( " tec " se conecta a la salida " detec" ) }
1 011 000 100 111101 000
1 100 000 111 111110 000
1 101 000 111 111110 000
1 110 000 111 111110 000
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla declara un decodificador de prioridad}
TTIN tec,c0,c1,c2,c3,c4,c5,c6,c7,p2,p1,p0;
TTOUT b5,b4,b3,p2,p1,p0;

1 ----- 001 --- 000 { si la señal tec es igual a uno, no se pregunta}
0 ----- 0 001 000 010 {por prioridad de la tecla y se regresa a la página }
0 ----- 01 001 001 010 { cero la cuenta y se pasa a la página siguiente}
0 ----- 011 001 010 010 { se obtiene en b5,b4 y b3 la información de la }
                                { columna }
0 --- 0111 001 011 010 { si se reciben 2 o más señales en bajo, se acepta }
0 -- 01111 001 100 010 { la de mayor prioridad }
0 - 011111 001 101 010
0 - 0111111 001 110 010
0 01111111 001 111 010
END TRUTH_TABLE

{ A partir de esta tabla, se deja activo en la página 010 el registro de captura, y se }
{ utilizan las variables de control a2,a1 y a0 para saber que bit es el que se esta }
{ transmitiendo}

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b0 }
TTIN a2,a1,a0,b0,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b0 en bn, se capturan las variables de control,}
000 - 010 ? ??? 011 {y se cambia a la página 011, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b1 }
TTIN a2,a1,a0,b1,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b1 en bn, se capturan las variables de control,}
001 - 010 ? ??? 100 { y se cambia a la página 100, para codificación sony }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b2 }
TTIN a2,a1,a0,b2,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
                                { Se copia b2 en bn, se capturan las variables de control,}
010 - 010 ? ??? 101 {y se cambia a la página 101, para codificación SONY}
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b3 }
TTIN a2,a1,a0,b3,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0; { Se copia b3 en bn, se capturan las variables de }

```

## Control Remoto de Formato Comercial

```

011 - 010 ? ??? 110 {control, y se cambia a la página 110, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b4 }
TTIN a2,a1,a0,b4,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
        { Se copia b4 en bn, se capturan las variables de control,}
100 - 010 ? ??? 111 {y se cambia a la página 111, para codificación }
END TRUTH_TABLE

BEGIN TRUTH_TABLE { Esta tabla tiene el control sobre el bit b5 }
TTIN a2,a1,a0,b5,p2,p1,p0;
TTOUT bn,a2,a1,a0,p2,p1,p0;
        { Se copia b5 en bn, se capturan las variables de control,}
101 - 010 ? ??? 100 {y se cambia a la página 100, para codificación }
END TRUTH_TABLE

{ El siguiente conjunto de tablas, son quienes codifican el bit bn en el formato: }
{ 3 espacios = inicio; 2 espacios = BAJO y 1 espacio = ALTO. }

BEGIN TRUTH_TABLE { Esta tabla manda la señal de inicio, y codifica el bit b0 }
        { en formato comercial }
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT a3,a2,a1,a0,out,p2,p1,p0;

0000 - 1 011 0001 0 011
0001 - 0 011 0010 0 011 { las variables de control a2,a1 y a0 se utilizan para}
0010 - 0 011 0011 0 011 { simular un registro de corrimiento y junto con a3 }
0011 - 0 011 0100 0 011 {garantizar que no haya combinaciones repetidas }
0100 - 0 011 0101 0 011
0101 - 0 011 0100 0 011 {se manejan 6 posiciones en cero, para lograr que el }
0100 - 0 011 0111 1 011 {espacio más pequeño, sea igual al tiempo de retorno}
0111 - 1 000 1000 1 011 { a uno. }
1000 0 1 011 1001 0 011
1001 0 0 011 1010 0 011 {si bn es cero, entonces se mandan 4 ceros y se }
1010 0 0 011 1011 0 011 { regresa a la página 010 para tomar el siguiente bit }
1011 0 0 011 1100 0 011
1100 0 0 011 0001 1 010
1000 1 1 011 1001 0 011 {si bn es uno, entonces se mandan 2 ceros y se }
1001 1 0 011 1010 0 011 { regresa a la página 010 para tomar el siguiente bit }
1010 1 0 011 0001 1 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b1 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0001 0 1 100 0 1010 100
1010 0 0 100 0 1011 100 {si bn es cero, entonces se mandan 4 ceros y se }
1011 0 0 100 0 1100 100 { regresa a la página 010 para tomar el siguiente bit }
1100 0 0 100 0 1101 100
1101 0 0 100 1 0010 010
0001 1 1 100 0 1010 100 {si bn es uno, entonces se mandan 2 ceros y se }
1010 1 0 100 0 1011 100 { regresa a la página 010 para tomar el siguiente bit }

```

```

1011 1 0 100 1 0010 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b2 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0010 0 1 101 0 1011 101
1011 0 0 101 0 1100 101 {si bn es cero, entonces se mandan 4 ceros y se }
1100 0 0 101 0 1101 101 { regresa a la página 010 para tomar el siguiente bit}
1101 0 0 101 0 1110 101
1110 0 0 101 1 0011 010
0010 1 1 101 0 1011 101 {si bn es uno, entonces se mandan 2 ceros y se }
1011 1 0 101 0 1100 101 { regresa a la página 010 para tomar el siguiente bit}
1100 1 0 101 1 0011 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b3 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0011 0 1 110 0 ? 1100 110
1100 0 0 110 0 ? 1101 110 {si bn es cero, entonces se mandan 4 ceros y se }
1101 0 0 110 0 ? 1110 110 { regresa a la página 010 para tomar el siguiente bit}
1110 0 0 110 0 ? 1111 110
1111 0 0 110 1 ? 0100 010
0011 1 1 110 0 ? 1100 110 {si bn es uno , entonces se mandan 2 ceros y se }
1100 1 0 110 0 ? 1101 110 { regresa a la página 010 para tomar el siguiente bit}
1101 1 0 110 1 ? 0100 010
END TRUTH_TABLE

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b4 en formato SONY}
TTIN a3,a2,a1,a0,bn,out,p2,p1,p0;
TTOUT out,a3,a2,a1,a0,p2,p1,p0;
0100 0 1 111 0 1101 111
1101 0 0 111 0 1110 111 {si bn es cero , entonces se mandan 4 ceros y se }
1110 0 0 111 0 1111 111 { regresa a la página 010 para tomar el siguiente bit }
1111 0 0 111 0 1000 111
1000 0 0 111 1 0101 010
0100 1 1 111 0 1101 111 {si bn es uno , entonces se mandan 2 ceros y se }
1101 1 0 111 0 1110 111 { regresa a la página 010 para tomar el siguiente bit }
1110 1 0 111 1 0101 010
END TRUTH_TABLE

{ Debido a que se requieren 12 pulsos de información, se recorren nuevamente los }
{ 5 primeros bits de información, así, se codifica solo una vez el bit b5 }

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit b5 en formato SONY}
TTIN a4,a3,a2,a1,a0,bn,out,p2,p1,p0; {se agrega a4 como indicador de recorrido}
TTOUT out,a4,a3,a2,a1,a0,p2,p1,p0;
        { si a4 se encuentra en cero, entonces se codifica el bit b5 y }
1 0101 - 1 100 0 1 0000 101 {se cambia de página para un segundo recorrido,}
0 0101 0 1 100 0 ? 1110 100 { Si a4 esta en uno, entonces se cambia a la }
        { página 101 sin transmitir b5}

```

```
0 1110 0 0 100 0 ? 1111 100 {si bn es cero , entonces se mandan 4 ceros y }
0 1111 0 0 100 0 ? 1000 100 { se regresa a la página 010 para tomar el sig. bit}
0 1000 0 0 100 0 ? 1001 100
0 1001 0 0 100 1 1 1000 010
0 0101 1 1 100 0 ? 1110 100 {si bn es uno , entonces se mandan 2 ceros y se }
0 1110 1 0 100 0 ? 1111 100 { regresa a la página 010 para tomar el siguiente bit}
0 1111 1 0 100 1 1 1000 010
END TRUTH_TABLE

{ Para el pulso número doce, se envía la información contenida en el bit Shift para}
{ duplicar el número de teclas }

BEGIN TRUTH_TABLE{ Esta tabla codifica el bit "Shift" bs en formato SONY}
TTIN a4,a3,a2,a1,a0,bs,out,p2,p1,p0; { Debido a ajuste de tiempos, y conside-}
TTOUT out,a4,a3,a2,a1,a0,p2,p1,p0; { randodo que este es el ultimo bit por tras-}
{ mititr, para este bit (bs) se hace una asignación menos a uno o a cero }
{ en la codificación comercial.}
1 0000 0 0 101 0 1 0001 101
1 0001 0 0 101 0 1 0011 101 {si bs es cero , entonces se mandan 3 ceros y se }
1 0011 0 0 101 0 1 0100 101 { regresa a la página 000 para permitir que otra }
1 0100 0 0 101 1 0 0000 000 { tecla sea aceptada }
1 0000 1 0 101 0 1 0001 101 {si bn es cero , entonces se mandan 4 ceros y se }
1 0001 1 0 101 1 0 0000 000 { regresa a la página 000 para permitir que otra }
END TRUTH_TABLE { tecla sea aceptada }
```

### Bibliografía

- [1] Programmable Logic Devices Databook and design guide. National Semiconductor.
- [2] OPAL TM, Manual de usuario.
- [3] Optoelectronics device data. Motorola.
- [4] Operational Amplifiers data book. National Semiconductor.