

## Reporte: Laboratorio #1

Esteban Rodríguez Quintana. B66076  
Entrega: 4/agosto/2022

### 1. Introducción

Este informe incluye el desarrollo del trabajo realizado durante el **laboratorio 1** del curso **Laboratorio de Microcontroladores**. Se abarca la creación de un circuito capaz de simular el funcionamiento de un bingo utilizando el microcontrolador PIC12f675 y otros componentes electrónicos necesarios para que se realice una integración hardware/software que cumpla con los requerimientos del diseño. Al finalizar este trabajo se logró implementar el diseño con las funcionalidades deseadas. Se concluye que la integración hardware/software es muy importante a lo largo de todas las etapas de desarrollo del proyecto, ya que debe de tomarse siempre en consideración su interfaz e interacción. Puede encontrar el código fuente y los archivos de simulación en el git: lab1\_microcontroladores

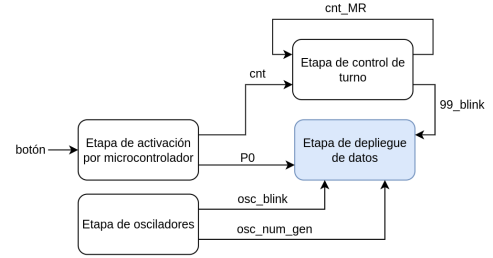


Figura 2: Diagrama de bloques funcionales

Como se describió anteriormente, 4 etapas que componen el diseño se enumeran y explican brevemente a continuación:

### 2. Nota teórica

Con la intención de crear un circuito capaz de simular el funcionamiento de un bingo de 16 turnos con números aleatorios, se planteó una solución compuesta de 4 etapas, las cuales se encargarán de realizar funciones específicas a nivel de hardware/software para cumplir con las especificaciones dadas. En las figuras 1 y 2 se muestran las etapas con sus respectivas señales de entrada y salida y además la interacción a nivel de interfaz entre ellas.

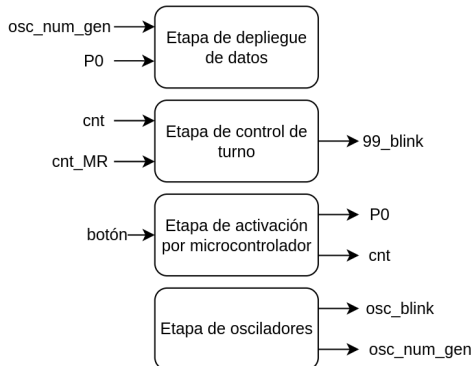


Figura 1: Diagrama de entrada/salida por etapa

1. Etapa de osciladores: Se muestra en la figura 3 y es la encargada de generar dos señales de reloj. Una de estas se utiliza para el parpadeo de los *display* en 99, mientras la otra se utiliza para generar una señal de entrada a uno de los módulos CD4026 que se emplea en la generación de número pseudo aleatorios.

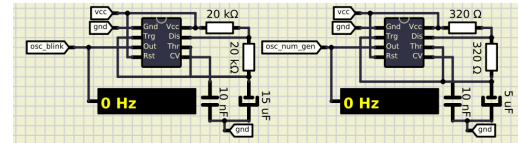


Figura 3: Etapa de osciladores

2. Etapa de control de turno: Esta se encarga del contar los turnos hasta llegar a 16 y reinicia el contador en el turno 17. También implementa la lógica con compuertas para activar el parpadeo de los *display*. Se muestra en la figura 4.

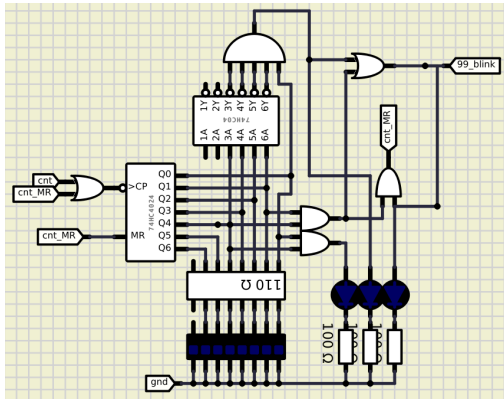


Figura 4: Etapa de control de turno

3. Etapa de despliegue de datos: Acá se muestran los números pseudo aleatorios en 2 *display* de 7 segmentos, además se emplean una serie de multiplexores con una señal de control proveniente de la etapa de control de turno. Esta etapa se muestra en la figura 5.

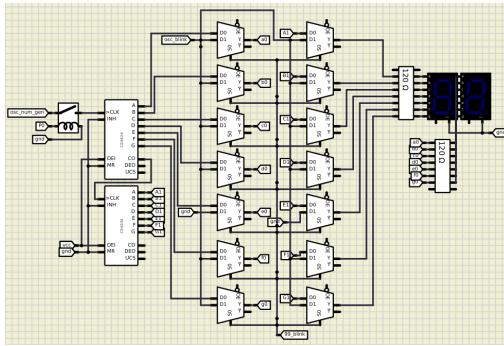


Figura 5: Etapa de display

4. Etapa de activación por microcontrolador: Mostrada en la figura 4, la pieza central de esta etapa es el PIC12f675. Este recibe una señal de activación mediante un botón y como salida da un pulso que se mantiene en 1 por un tiempo pseudo aleatorio asignado mediante un contador por software.

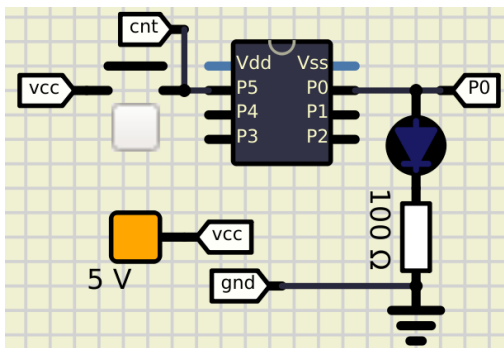


Figura 6: Etapa de display

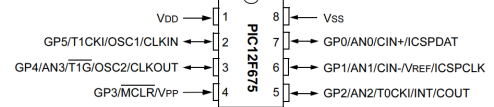


Figura 7: Pines PIC12f675

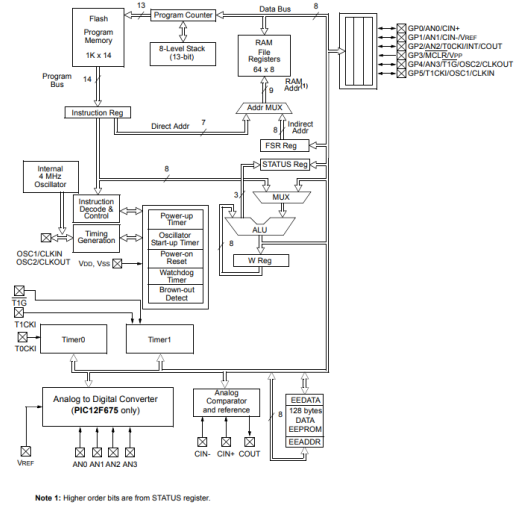


Figura 8: Registro TRISIO

## 2.1. Componentes

Para el diseño implementado se emplearon un conjunto de compuertas lógicas, componentes pasivos y el microcontrolador PIC12f675, como se describe a continuación.

### 2.1.1. Microcontrolador PIC12f675

El PIC12675 es un microcontrolador de 8 bits, 8 pines y basado en FLASH. Además utiliza arquitectura RISC y cuenta con un set de solamente 35 instrucciones. Tiene la capacidad de trabajar con oscilador interno de 4 MHz o externo empleando un cristal oscilador y resonadores. Su voltaje de operación ca desde -2 V hasta 5.5 V. Además cuenta con 1024 palabras en su memoria para programa y un total de 192 bytes distribuidos entre memoria SRAM y EEPROM. Este microcontrolador cuenta con 6 entradas/salidas con diferentes funcionalidad y posibles configuraciones, como se muestra en 7. Puede consultar la hoja de datos disponible en la página web del fabricante: PIC12F629/675 Data Sheet. En la figura 8 se observa el diagrama de bloques del microcontrolador.

La programación del PIC12f675 se configuraron los registros de función especial TRISIO (figura 9)y GPIO (figura 10). El registro GPIO es el que se utiliza para asignar valores o leer las salidas/entradas de los puertos GPIO[0-5] que a su vez se configuran como I/O utilizando del registro TRISIO.

REGISTER 3-2: **TRISIO — GPIO TRISTATE REGISTER (ADDRESS: 85h)**

U-0	U-0	R/W-x	R/W-x	R-1	R/W-x	R/W-x	R/W-x
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0
bit 7		bit 0					

bit 7-6: **Unimplemented:** Read as '0'

bit 5-0: **TRISIO<5:0>**: General Purpose I/O Tri-State Control bit  
1 = GPIO pin configured as an input (tri-stated)  
0 = GPIO pin configured as an output.  
**Note:** TRISIO<3> always reads 1.

Figura 9: Registro TRISIO

REGISTER 3-1: **GPIO — GPIO REGISTER (ADDRESS: 05h)**

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0
bit 7		bit 0					

bit 7-6: **Unimplemented:** Read as '0'

bit 5-0: **GPIO<5:0>**: General Purpose I/O pin.  
1 = Port pin is >V<sub>IH</sub>  
0 = Port pin is <V<sub>IL</sub>

Figura 10: Registro GPIO

## 2.2. Componentes electrónicos complementarios

En cuanto a los componentes complementarios, en las etapas de control de turno, *display*, osciladores y activación por microcontrolador se emplearon los componentes descritos y encapsulados mostrados en la tabla 1 que, dado a su tamaño, podrá encontrar en la sección de apéndices, ahí hay una breve descripción de su papel dentro del diseño. Adicionalmente, en los apéndices se encuentran links hacia las hojas de datos de dichos componentes para la revisión de sus características de operación/funcionamiento.

## 2.3. Justificaciones de diseño

En la tabla 1 puede encontrar explícitamente la función de cada componente dentro de su etapa, excepto del microcontrolador, al cual ya se le dedica toda una sub sección. En términos de justificar los valores seleccionados para los componentes, especialmente pasivos, en el caso de las conexiones hacia LEDs se procuró colocar resistencias de al menos 100  $\omega$  para que ante el mayor voltaje disponible, 5 V, la corriente no supere los 50 mA, esta decisión se basa en las especificaciones disponibles en las hojas de datos. Por su parte, para la etapa de osciladores, se empleó un diseño estándar de oscilador monoestable disponible en la hoja de datos del lm555 y se iteró sobre los valores de las resistencias y el capacitor electrolítico para lograr las frecuencias de oscilación deseadas. Para encontrar dichas frecuencias de operación, se emplearon los datos disponibles en las curvas mostradas en la figura 11.

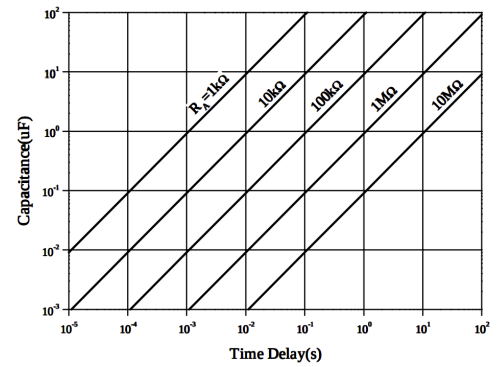


Figura 11: Curvas de delay lm555

### 2.3.1. BOM

En la tabla 2 se muestran los componentes utilizados y una breve descripción de su función dentro del diseño, su precio y cantidad empleada. Esta tabla la podrá encontrar en la sección de apéndices

## 3. Desarrollo

En la integración de las 4 etapas propuestas en el diseño se destaca que la funcionalidad del bingo es la adecuada y se adapta a los requerimientos planteados por el enunciado de laboratorio:

1. Desarrollara un simulador de tómbola simplificada de bingo como el de la figura 1 utilizando dos displays de 7 segmentos para formar la pantalla, un botón, el microcontrolador PIC12f675 y cualquier otro componente que considere necesario.
2. Cada vez que se presiona el botón debe desplegarse en el display un numero que represente a una de las bolas de la tómbola que van del 00 al 99. Para efectos de simplificar la simulación considere que la tómbola solo tendrá 16 bolas.
3. Después de sacar 16 bolas/números deberá parpadear la pantalla con el numero 99 y reiniciar el juego.

En cuanto al funcionamiento general del diseño, en la figura 12 se muestra el circuito completo con sus 4 etapas y en el turno 0, en el cual la pantalla muestra el 99 parpadeando y además de los LED indicadores en la derecha de la etapa de control encendidos, indicando en inicio del juego. Además, note que en esta misma etapa se muestra el LED bar con el equivalente a 1 en binario, cuando este LED bar se pone en 2 binario es cuando inicia el primer turno y llega hasta 17 binario antes de reiniciarse a 0 para empezar nuevamente el juego. Puede observarse en la etapa de osciladores que las frecuencias del oscilador de parpadeo y de generación de números son 1.6 Hz y 304 Hz, correspondientemente. El juego empieza en *parpadeo de 99*, al presionar el botón, se pasa al estado de *juego* durante 16 turnos para finalmente llegar al estado de *reset* donde el contados de la etapa de control de turno vuelva a 0.

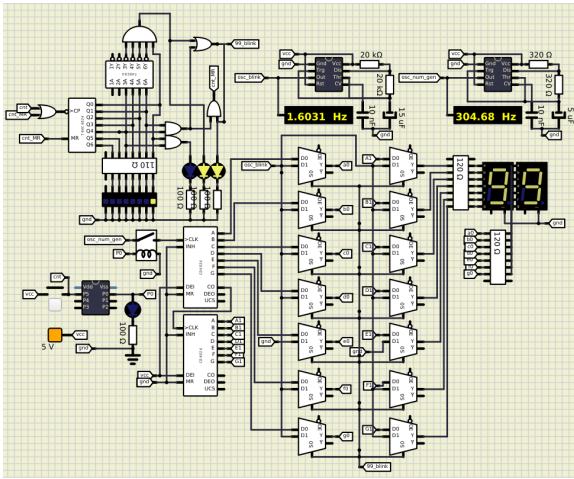


Figura 12: Diseño completo

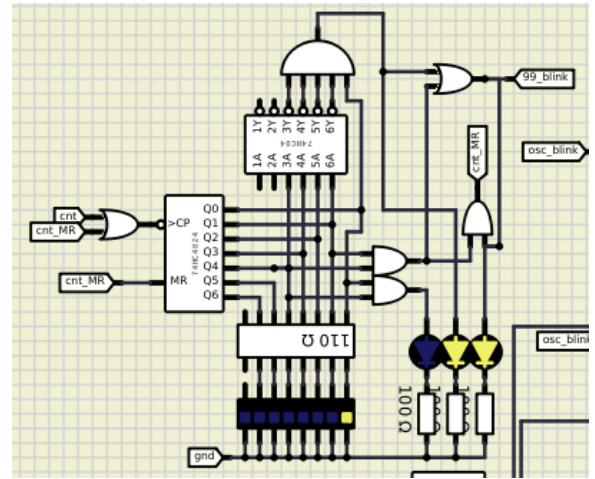


Figura 13: Turno 0

La siguiente lista describe señales importantes en el diseño y como estas son obtenidas:

1. *osc.blink*: es una señal lenta encargada de dirigir la salida el oscilador de parpadeo hacia la etapa de *display*, donde se conecta a la entrada de 6 de los 7 flip flops empleados por cada *display* de 7 segmentos para generar un 99 parpadeante.
2. *osc.num\_gen*: Esta señal de reloj es rápida y tiene la finalidad de generar pulsos que, ante la activación del botón en la etapa de activación por microcontrolador, lleguen hasta la entrada CLK del módulo DC4026, que además se mantiene pasando a través de un relay activado por el pin P0 del PIC12f675.
3. *cnt*: Esta se pone en alto cada vez que le botón en la etapa de activación por microcontrolador se presiona. Su función es generar un pulso que llegue al módulo 74HC4024 y que, en una OR con la señal *cnt\_MR* para incrementar la cuenta en cada turno desde 1 hasta 17.
4. *P0*: esta señal sale de la etapa de activación por microcontrolador hacia la etapa de *display*. Esta salida de PIC12f675 se asigna mediante software de tal manera que se pone en 1 cada vez que se lee un 1 proveniente de *cnt/P5*, dicha salida se pone en alto durante un tiempo pseudo aleatorio que se obtiene usando un contador dentro del software.
5. *cnt\_MR*: Señal encargada de reiniciar el contador de turno. Esta se pone en estado activo cuando se llega a 18 binario y además se está en el estado de parpadeo 99.
6. *99.blink*: Esta señal permite que los multiplexores en la etapa de *display* activen su entrada D1, dejando pasar los 99 hasta llegar a los display de 7 segmentos. Esta señal se genera cuando el contador 74HC4024 está en 1 binario o cuando se ha presionado el botón 17 veces.

La figura 13 muestra la etapa de control de turno durante el *parpadeo 99*, como se observa, el LED bar muestra 1 en binario y además 2 de los 3 los LED indicadores de la derecha.

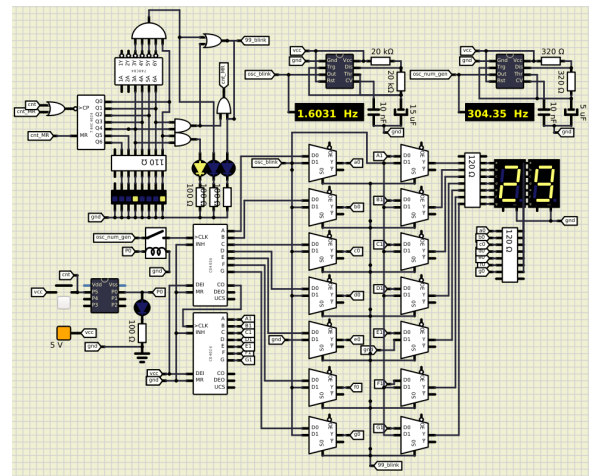


Figura 14: Turno 16

En la figura 14 se puede ver el diseño completo para el turno 16. Vale la pena recalcar que como el contador de turno se está iniciar siempre en 1, hay un desfaz de entre el número de turno y el LED bar en binario, donde el número de turno va 1 número por delante del mostrado en el LED bar. Por lo anterior es que en el turno 16 el LED bar muestra un 17 binario.

A nivel de software, se empleó una idea sencilla la cual consiste en utilizar un contador y aprovechar la frecuencia de actualización de los datos en el PIC durante un *while* infinito para ayudar en la generación del número a mostrar en los *display*. Lo que se hace es incrementar un contador en cada ciclo y en caso de que se reciba una señal alta en el pin P5, entonces se pone un 1 el pin P0 durante el tiempo que indique la variable de dicho contador. El código puede observarlo a continuación:

```
1 #include <pic14/pic12f675.h>
2
3 void delay(unsigned int time);
4
5 // Main funct
6 void main(void)
7 {
8     TRISIO = 0x00;           // Set all output
```

```

9  GPIO = 0x00;          // Make all pins 0
10
11  TRISIO5 = 1;          // Make GP5 as input
12  unsigned int time = 100;
13  unsigned int random = 0x00;
14  while(1)
15  { random += 0x01; // keep += 1
16    if(GP5){ // if read GP5 in as 1
17      GP0 = 0x01;
18      delay(random);
19      GP0 = 0x00;
20    }
21  }
22 }
23 // delay func
24 void delay (unsigned int time ){
25   unsigned int i ;
26   unsigned int j ;
27   for( i =0; i <time ; i ++ )
28   for( j =0; j <1275; j ++ );
29 }

```

Listing 1: Código para bingo

### 3.1. Análisis de resultados

Tras realizar toda la explicación de la implementación y diseño, en esta sección se abarcan brevemente los resultados obtenidos. En las figuras 15 y 17 se muestran las formas de onda de las señales provenientes de la etapa de osciladores y podemos observar como la señal de Out, para ambos lm555 conmuta a través del tiempo.

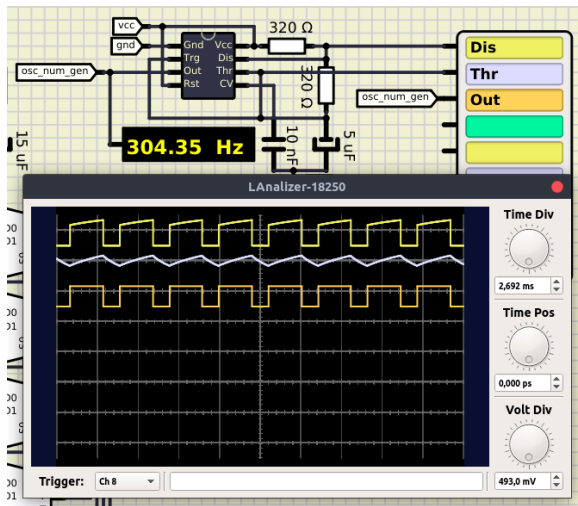


Figura 15: Señales de oscilador de generación de números

La intención de ajustar las resistencias y capacitor electrolítico de los osciladores fue generar una señal lenta que permitiera el parpadeo de los *display* cada más o menos 1.6 segundos y otra señal bastante rápida que conmutara bastante en un periodo pequeño de tiempo para aportar aleatoriedad al número a mostrar. Por otro lado, la aleatoriedad de los datos es en realidad pseudo aleatorio, ya que no se cuenta con un proceso estocástico del cual obtener aleatoriedad absoluta. Sin embargo, la generación de los números es bastante aleatoria en el sentido de que, hay oscilación disponible rápida en la etapa de oscilación, adicionalmente, esto se conjuga con la

velocidad de cambio del contador dentro del firmware del pic y su velocidad de operación. Por lo anterior, el usuario nunca podrá tener noción real de cual número será el siguiente dentro del juego.

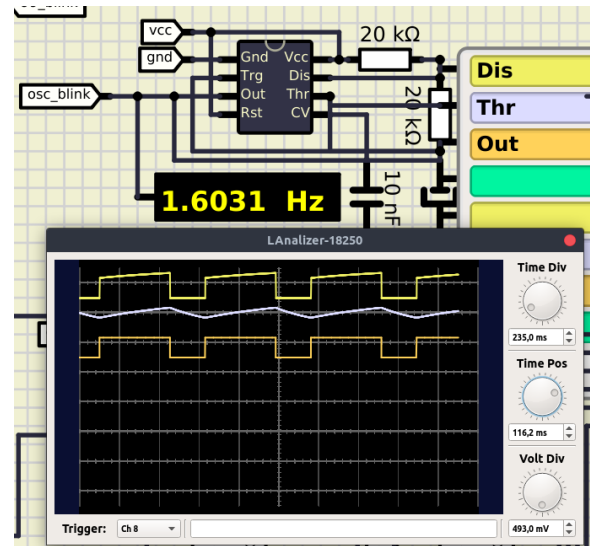


Figura 16: Señales de oscilador de parpadeo

Por su parte, como muestra de los voltajes presentes en los LED y sus resistencias de protección, vemos que según la figura ??, la tensión a través del LED es 2.41 V y a través de la resistencia es de 1.85 V, por tanto la corriente que atraviesa estos componentes es de 18 mA. Dicha corriente está por debajo el límite de 50 mA propuesto en la nota teórica. En el caso de la imagen 18, podemos obtener que la tensión a través de la resistencia sería de 1.93 V y que a su vez, la corriente resultante a través del conjunto LED/resistencia es de 16 mA, cumpliendo también el resultado propuesto en la nota teórica.

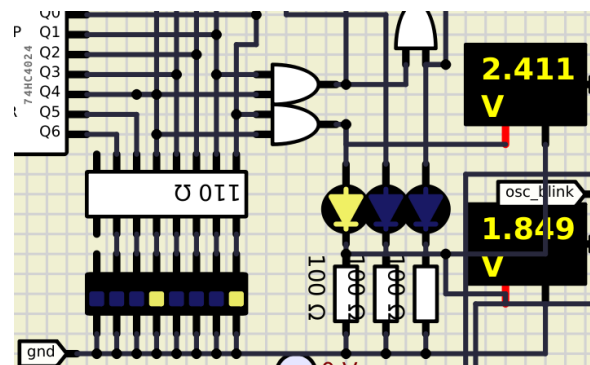


Figura 17: Tensiones de LED en etapa de control de turno



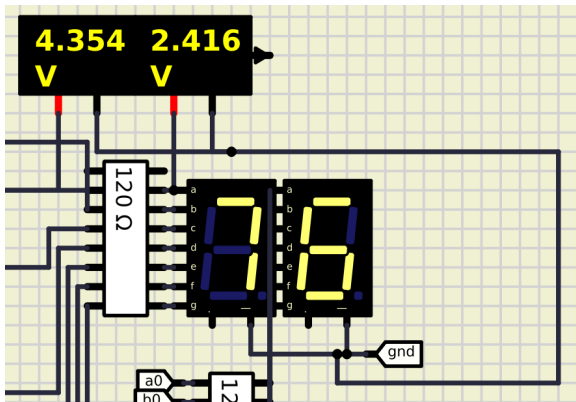


Figura 18: Tensiones de LED en displa de 7 segmentos en etapa de *display*

## 4. Conclusiones y recomendaciones

- Es importante contemplar la integración de las etapas diseñadas y además la integración entre el software y hardware para que los resultados del diseño completo sean apropiados y se ajusten a los requerimientos.
- La generación de un número completamente aleatorio puede requerir de fuentes que a nivel de simulación no son tan fáciles de obtener y por tanto, se puede contextualizar el problema ante la percepción del usuario respecto a la velocidad de las señales. Permitiendo que el número obtenido en realidad si sea aleatorio en términos de la capacidad del usuario para este sistema cerrado.
- Se logra cumplir con todos los requerimientos conductuales del diseño y con la aleatoriedad de los datos basado en distintas fuentes de señal.
- El diseño puede quedar limitado a nivel de hardware si se deseara implementar una versión con más turnas, ya que esto implicaría modificar la lógica de control de turno. Sin embargo, Ante la falta de recursos en la etapa de activación por microcontrolador, la solución propuesta se basa mayor mente en hardware para poder disminuir estas deficiencias a nivel de controlador y podría ser además muy útil si se desea implementar alguna otra funcionalidad a nivel de software.
- Únicamente se utilizan dos entradas/salidas del microcontrolador para todo el proyecto, por tanto, se pueden tener recursos libres a ese nivel para implementar otras funcionalidades.
- El costo económico podría haber sido menor si se empleaba una solución más enfocada hacia el software.

## Referencias

Se consultaron hojas de datos referidas en las sección de apéndices.

## 5. Apéndices

Componente	Descripción	Etapas	Función en diseño
CD4026	IC Counter/Divider decade	Display	Es un contador que recibe una señal de entrada CLK y funciona como un contador con memoria interna que es capaz de convertir dicha cuenta a salidas que concuerdan con las entradas de los displays de 7 segmentos para poder mostrar los números deseados. Se empleó un dispositivo por cada display y se encadenaron de manera que uno brinda unidades y otro decenas. Las salidas de estos se encuentran conectadas a una etapa de flip flops que se encarga de seleccionar entre hacer un blink en 99 o el número generado de manera pseudo aleatoria.
74HC402X	Contador binario de 7 etapas con ripple carry	Control de turno	Este módulo tiene la capacidad de contar desde 0 hasta 127 y se emplea para contar los turnos desde 0 a 16. En el turno 0 siempre se muestra el 99 parpadeando. Cuenta con una señal de MR (master reset) para poner el contador en 0.
74HC04	Compuerta inversora 6 entradas	Control de turno	módulo de inversores miscelaneo para la lógica de parpadeo.
LM555	Temporizador	Osciladores	Este se utiliza para generar 2 señales de reloj independientes. Una de estas señales es de ~1.6 Hz y se usa en el parpadeo 99, mientras la otra señal es de ~304 Hz y se usa como parte el proceso de generación de los números pseudo aleatorios.
XDMR14C	Display de 7 segmentos	Display	2 pantallas de 7 segmentos para desplegar números entre 0 y 9 cada una.
Resistencias	-	varias	Resistencias de varios valores para protección de componentes y regulación tensión/corriente
Capacitores	-	Osciladores	Empleados como parte del circuito de oscilación
Compuertas lógicas discretas	AND/NAND/OR	Control de turno	Estas se utilizan par generar lógica que permita controlar las señales de MR y de habilitación de parpadeo.
LED discretos	indicadores de estado/acción	varios	Estos se utilizaron como indicadores
LED bar	Barra de LED	Control de turno	Esta barra de LED se usa para visualizar el estado del programa de forma más clara, ya que permite visualizar en binario en turno en que se encuentra el juego de bingo. Es importante mencionar que el turno 0 es cuando los LED parpadean en 99.
Relay	Relay para activación	Activación por microcontrolador	activar la señal de cnt y P0 en alto

Cuadro 1: Tabla de componentes y función

Componente	Cantidad	Precio unitario	Precio Total	Descripción
PIC12f675	1	\$1,80	\$1,80	Microcontrolador
CD4026	2	\$0,60	\$1,20	IC Counter/Divider decade
74HC4026	1	\$1,25	\$1,25	Contador binario de 7 etapas con ripple carry
74HC04	1	\$0,37	\$0,37	Compuerta inversora 6 entradas
LM555	2	\$2,06	\$4,12	Temporizador
XDMR14C	2	\$0,91	\$1,82	Display de 7 segmentos
4116R-1-XXXXX	3	\$1,66	\$4,98	Arreglo de 8 resistencias
Resistencias discretas	5	\$0,30	\$1,50	Resistencias de varios valores
Capacitores discretos	4	\$0,20	\$0,80	Capacitors cerámicos y electrolíticos
LED discretos	4	\$0,80	\$3,20	LED discretos para visualización de etapas
LED bar	1	\$6,70	\$6,70	Barra de LED
relay	1	\$2,00	2	relay de control

Cuadro 2: Bill of Materials

Puede hacer click sobre los siguientes nombres de componentes para abrir las hojas de datos correspondientes:

1. PIC12f675
2. CD4026
3. 74HC4026
4. 74HC04
5. LM555
6. XDMR14C
7. 4116R-1-XXXXX
8. Resistencias discretas
9. Capacitores discretos
10. LED discretos
11. LED bar
12. relay