

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

III ciclo 2021

Reporte de Laboratorio #2

GPIOs, Timers y FSM

Edward Cerdas Rodríguez - B71956

Grupo 01

Profesor: MSc. Marco Villalta Fallas

22 de Enero de 2021

1. Introducción/Resumen

Los > microcontroladores son dispositivos muy útiles con los cuales se puede manipular corrientes que fluyen por distintas partes de los circuitos mediante la programación de su firmware, esto los hace muy versátiles, ya que con un mismo modelo de microcontrolador se pueden crear diferentes circuitos controlados según se necesite.

Para este laboratorio, el microcontrolador a utilizar será un ATtiny4313, el cual es bastante sencillo, sin embargo es un poco más versátil que el utilizado en el laboratorio anterior, dado que en este caso tenemos más pines con los que podemos jugar a la hora de crear proyectos. Se pretende modificar el firmware para simular un juego de Simón dice, el cual consiste en observar una secuencia en 4 leds y luego replicar este patrón presionando los botones en el orden correcto. Cada led estará ligado a un botón, para un total de 4 leds y 4 botones. Cada uno de los leds se encuentra conectado a un pin que fue previamente configurado como salida en el respectivo registro. Además, cada uno de los botones está conectado a un pin que fue configurado como entrada, y además, se tomaron otras consideraciones, tales como el puerto al que pertenece el pin y otros que serán mencionados a lo largo del reporte de laboratorio.

La idea principal es simple, el microcontrolador muestra una secuencia random de 4 luces para iniciar y el jugador debe replicar la secuencia mostrada, si la secuencia es errónea, el programa finaliza, pero si la secuencia es correcta, se continúa al siguiente nivel, donde se mostrarán 5 luces, si se vuelve a acertar ahora serán 6 luces y así, agregando un led por nivel, sucesivamente, hasta que el jugador se equivoque en la secuencia.

Otra de las cosas que se pretende, es que cada una de las rondas sea más difícil que la anterior, no solo por el hecho de que será un led más para cada nivel, sino que para cada nivel, el tiempo que cada led se encontrará encendido será de 200ms menos que el nivel anterior, en el primer nivel, el tiempo que dura encendido el led es de 2 segundos. Esto se llevará a cabo por medio de interrupciones con los timers propios del microcontrolador.

En general, se busca explorar opciones sobre lo que puede y no puede llegar a hacer un microcontrolador de estas características, de este modo, comprender de una mejor manera el funcionamiento de los GPIOs y las interrupciones.

2. Nota Teórica

En esta sección se va a incluir la información del microcontrolador, periféricos utilizados, componentes electrónicos complementarios; así como también el diseño del circuito.

2.1. Características generales ATtiny4313

El microcontrolador ATtiny4313 es un dispositivo que cuenta con 20 pines, los cuales están divididos en 3 puertos (A, B y D), sin embargo, solamente 18 de ellas pueden ser utilizadas para funciones de I/O, debido a que las otras dos son pines de VCC y de GND, tal como se muestra en la figura 1.

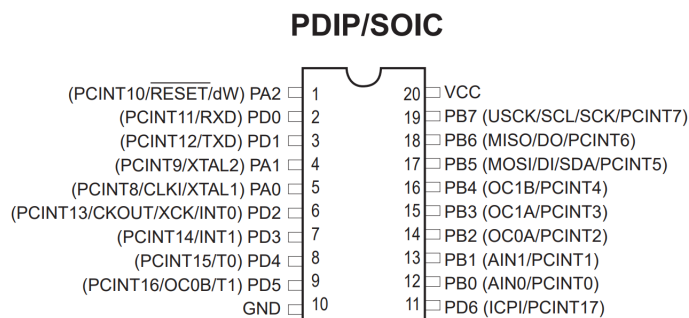


Figura 1: Esquema del microcontrolador [1]

Algunas de las principales características de este microcontrolador, tomadas de la hoja del fabricante, son:

- 18 pines de entrada y salida.
- Microcontrolador de 8 bits.
- 120 instrucciones.

- 3 diferentes puertos.
- 32 registros.
- Interrupciones internas y externas.
- 4 canales PWM.
- Temporizadores 8 bits.
- 256 Bytes de SRAM interna.

2.2. Características eléctricas

Las características eléctricas de este microcontrolador, tales como las corrientes y tensiones de entrada y salida de los pines se muestran en la figura 2

22. Electrical Characteristics

22.1 Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0 mA
DC Current V_{CC} and GND Pins.....	200.0 mA

Figura 2: Características eléctricas del microcontrolador [1]

2.3. Periféricos utilizados/descripción de registros e instrucciones

Los registros más importantes utilizados en este laboratorio son:

2.3.1. PORTx

En este caso, el PORTx es la base del laboratorio, ya que es lo que nos permite cambiar de estado las salidas dependiendo de los estados de las entradas. Las 4 leds conectados a 4 pines diferentes, los cuales están programados como salidas, están conectadas en el puerto B, por lo que, realmente solamente se utiliza el registro PORTB para este laboratorio.

10.3.5 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x18 (0x38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 3: PORTB [1]

2.3.2. DDRx

Tal y como se comentó anteriormente, se requería que algunos de los pines funcionen como salidas, mientras que otros como entradas. En este caso, el registro DDRB es el que debe ser modificado para lograr esto. De esta forma, se configuraron los pines B2, B3, B4 y B5 como salidas poniendo un 1 en el bit respectivo. Para los pines que fueron configurados como entrada se pone un 0 en el bit respectivo.

10.3.6 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 4: DDRB [1]

2.3.3. GIMSK

Este registro funciona para habilitar las interrupciones, tales como las interrupciones externas (INT0 e INT1) o las interrupciones por cambio en el pin (PCINT0, PCINT1 y PCINT2)

GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	INT1	INT0	PCIE0	PCIE2	PCIE1	–	–	–	GIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Figura 5: GIMSK [1]

2.3.4. PCMSKx

En este registro se habilitan las interrupciones para el pin específico, ya que, por ejemplo, si se quiere utilizar el pin A0 como interrupción por cambio en el pin, se debe levantar el bit correspondiente, de otro modo, no se reconocerá con solo cambiar el registro GIMSK.

Para este laboratorio, los utilizados fueron tanto PCMSK como PCMSK1, los cuales son necesarios para utilizar las interrupciones en los pines B0 y A0 respectivamente.

PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
0x20 (0x40)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 6: PCMSK [1]

PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	–	–	–	–	–	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 7: PCMSK1 [1]

2.3.5. SREG

Este es un registro de 8 bits, en el que el bit 7 debe ser puesto en alto para habilitar la interrupción global, lo cual significa que podremos utilizar las interrupciones del microcontrolador.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 8: PCMSK1 [1]

2.3.6. TCCR0A

En este registro se debe seleccionar el modo de funcionamiento del contador, que en este caso será normal, ya que nos interesa que cuente de 1 en 1 desde 0 hasta 255.

TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 9: TCCR0A [1]

2.3.7. TCNT0

Este registro se utiliza para guardar el valor del contador, este registro cuenta con 8 bits, por lo que el mayor número que puede guardar es 255. Es por esto que se cuenta únicamente de 0 a 255.

TCNT0 – Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
0x32 (0x52)	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 10: TCNT0 [1]

2.3.8. TIMSK

Por su parte, TIMSK es un registro análogo al GIMSK, pero en vez de habilitar las interrupciones externas o por cambio de pines, sirve para poder activar las interrupciones de Timer0 y Timer1.

TIMSK – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x39 (0x59)	TOIE1	OCIE1A	OCIE1B	–	ICIE1	OCIE0B	TOIE0	OCIE0A	TIMSK
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 11: TIMSK [1]

2.3.9. TCCR0B

Este es un registro que sirve para crear un re-escalado de la frecuencia de operación del microcontrolador, ya que, a ciencia cierta, frecuencias de MHz son demasiado altas como para ser controladas fácilmente por un humano. En pocas palabras, este registro modifica la frecuencia a la que trabaja el timer, a partir de la frecuencia original, dividiéndola en una fracción mucho menor que la inicial.

TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 12: TCCR0B [1]

2.4. Componentes elegidos y precios

- 1 microcontrolador ATtiny4313 - precio aproximado de 800 colones cada uno.
- 4 resistencias de 180Ω- precio aproximado 29 colones cada una.
- 4 resistencias de 4700Ω- precio aproximado 59 colones cada una.
- 4 botones - precio aproximado 90 colones cada uno.
- 4 leds de (uno rojo, uno azul, uno verde y uno amarillo) - precio aproximado entre 130 y 370 colones cada una dependiendo del color.
- 1 batería de 5 volts o fuente en caso de tener disponible - precio aproximado 1000 colones cada batería.
- 4 capacitores de 1uF - precio aproximado 85 colones cada uno.

2.5. Justificación del diseño del circuito con los componentes elegidos

En este caso para el diseño se utilizaron componentes realmente básicos.

Los 4 leds que se utilizaron necesitaban tener una resistencia en serie para que no se quemen, tomando en cuenta que el pin puede entregar 5 volts para alimentar cada uno de los leds, es conveniente utilizar una resistencia de por lo menos 180 Ω, esto dará un brillo alto, alimentando a los leds con una corriente de aproximadamente $I = \frac{V}{R} = \frac{5}{180} \approx 27\text{mA}$. Sin embargo, se pueden elegir valores un poco más altos para atenuar el brillo o en caso de que no se quiera entregar una corriente tan alta, por ejemplo, se pueden utilizar resistencias de 270 Ω, las cuales limitarán la corriente a aproximadamente 18,5mA. Para este laboratorio se eligen las resistencias de 180Ω ya que en el simulador, permite ver, de una mejor manera, el brillo de los leds.

Además, otro de los componentes completamente necesarios son los botones, los cuales se utilizan para generar la entrada que es detectada en el pin del microcontrolador. Estos son alimentados con 5 volts, por lo que se requiere el uso de resistencias de suficiente tamaño para que a los pines del microcontrolador no les llegue más corriente de la que pueden soportar. En este caso se utilizan resistencias de 4.7kΩ las cuales limitarán la corriente que llega al pin a tan solo 1mA. Acá solamente se menciona por el fabricante que se pueden utilizar corrientes de entrada al pin de hasta unos 20mA, siempre y cuando no se superen los 60mA por puerto, ya que superando esas condiciones no se aseguran resultados confiables.

Seguidamente, la batería o fuente de 5V, se utiliza debido a que se quiere generar un cambio en la entrada de los pines configurados como entrada. La forma en la que se planteó resolver esto es conectando una entrada de 5V fija a cada uno de los pines, pero que cuando se presione el botón, el pin se conecte directo a tierra, bajando el valor desde los 5V hasta los 0V.

Por último, los capacitores utilizados de 1uF cada uno, cumplen una función que en el simulador realmente no es un problema, pero en la vida real sí existe y es que los pulsadores tienen problemas mecánicos que conllevan a rebotes en la señal, esto puede causar problemas en el funcionamiento del circuito si no se toman medidas ya sea por software o por hardware. En este caso se decidió optar por un capacitor en paralelo con el pulsador, el cual

va a hacer que la señal tome la forma de la carga del capacitor, haciendo que la señal no tenga saltos de 0 a 1 en diferentes instantes de tiempo.

El valor del capacitor fue elegido tomando en cuenta el tiempo τ , recordando, que para una carga de aproximadamente el 100 % de un capacitor se necesitan 5τ . El valor de la constante τ se puede calcular de la siguiente forma:

$$\tau = R \cdot C = 4700\Omega \cdot 1\mu F = 0,0047s \quad (1)$$

De lo anterior sabemos entonces que el capacitor tarda aproximadamente 0,0235s en cargarse por completo. Este tiempo es suficiente para que no haya problemas de rebote en la señal de entrada al pin 5 y lo suficientemente pequeño como para evitar otros problemas que se pueden tener si el valor del capacitor es demasiado grande, tales como falsas lecturas debido a que el capacitor aún se encuentra cargando o descargando.

3. Desarrollo/Análisis de resultados

Para el desarrollo del laboratorio se utilizó el software de SimulIDE para crear el esquemático del circuito, el cual se muestra en la figura 13.

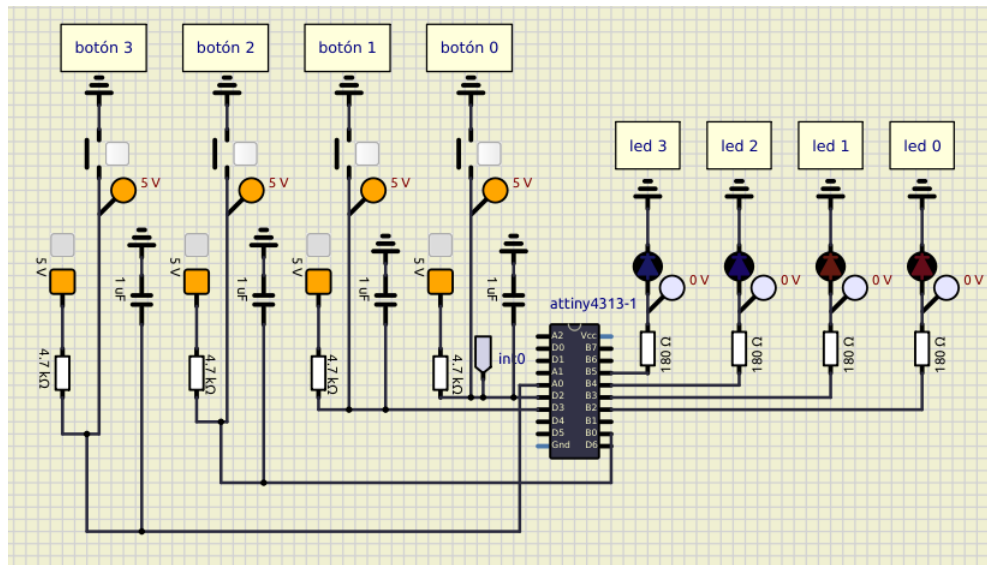


Figura 13: Circuito creado para la simulación [Elaboración propia]

El funcionamiento general del circuito puede ser descrito de la siguiente forma: Los pines B2, B3, B4 y B5 funcionan como salidas, las cuales son las encargadas de encender y apagar los leds según corresponda. Además, Los pines B0, A0, D2 y D3 funcionan a manera de entrada, no todos funcionan de la misma manera, sin embargo, se realizó la configuración correspondiente para cada uno de ellos, de modo que todo funcione bien en conjunto. Para este fin, se utilizaron registros como GIMSK, PCMSKx y DDRx.

Una vez se detecta el cambio en alguno de los pines configurados como entrada, el programa empieza, mostrando una secuencia de 4 luces led generada por números random, el usuario debe replicar esta secuencia presionando los botones en el orden correcto, si acierta, continúa con la siguiente ronda, en donde la cantidad de led que se encienden aumenta en 1, si falla, el juego termina.

Mostrar el funcionamiento de este programa en imágenes es algo difícil, ya que es bastante interactivo y tomar capturas de pantalla del momento preciso es un poco complicado, sin embargo, a continuación se muestran un par de imágenes de algunos de los leds siendo encendidos:

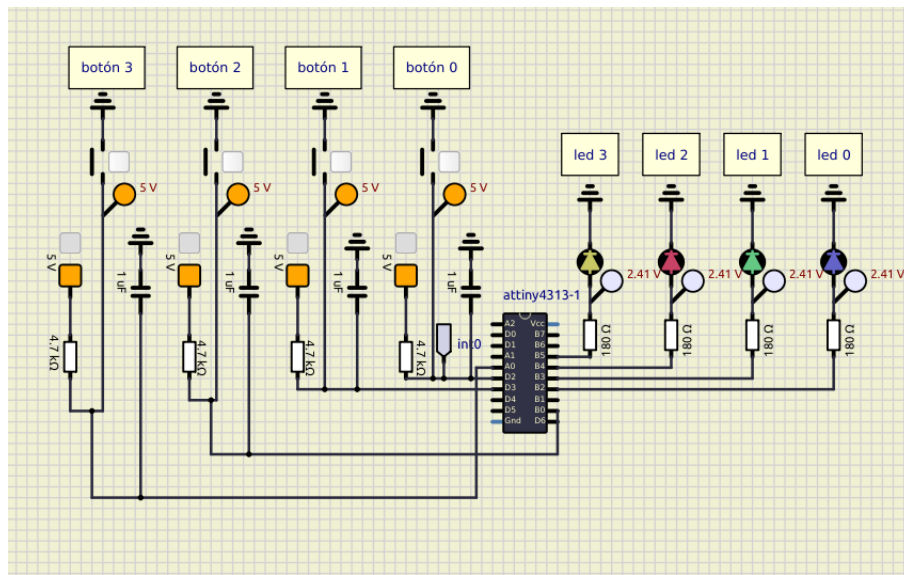


Figura 14: Inicio del programa [Elaboración propia]

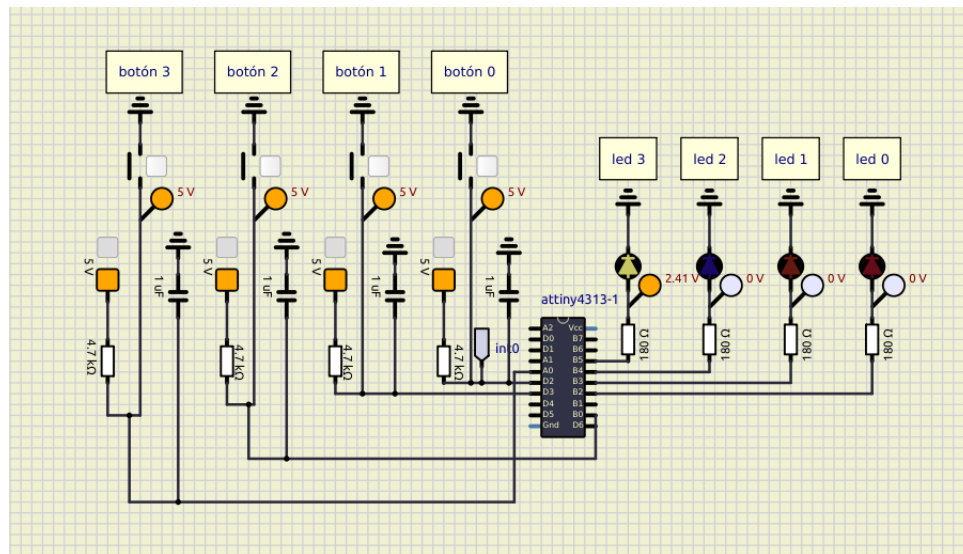


Figura 15: Juego encendiendo el led número 3 como parte de la secuencia [Elaboración propia]

3.1. Diagrama de bloques

El diagrama de bloques del funcionamiento de la máquina de estados programada para este juego de Simón dice se muestra en la figura 16.

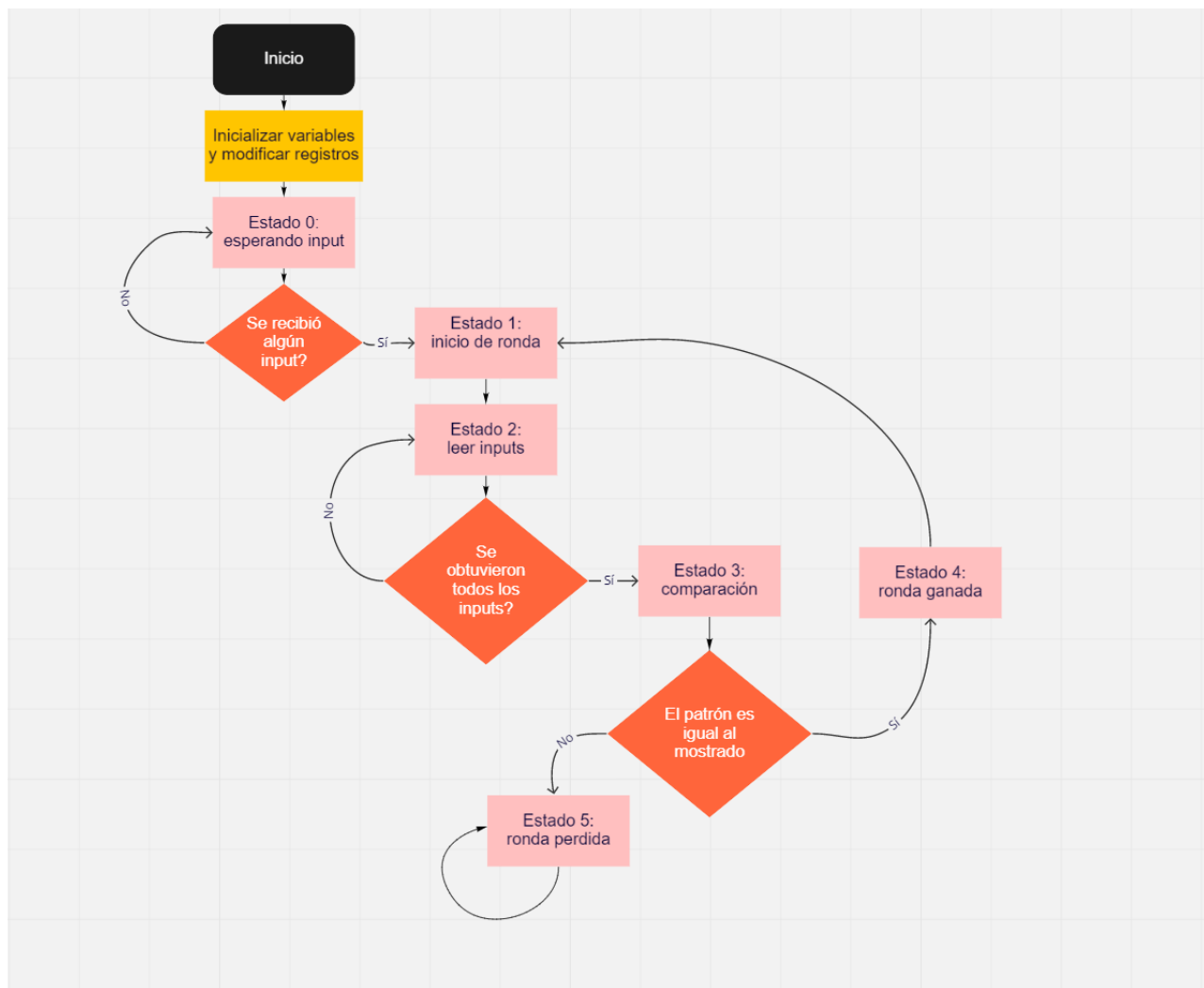


Figura 16: Diagrama de bloques de la máquina de estados [Elaboración propia]

4. Conclusiones y recomendaciones

- Se logró entender de una mejor forma el funcionamiento de las interrupciones y de las FSM.
- Se determinó el funcionamiento general de microcontrolador ATtiny4313 sus registros y métodos de operación.
- Se logró comprender conceptos muy importantes para la utilización de los microcontroladores, tales como interrupciones, estados y registros.
- Se entendió de mejor manera la generación de números aleatorios utilizando contadores en loops infinitos.

Recomendaciones: En el caso de este proyecto, se pretendía usar los timers del microcontrolador para la temporización de leds, lo cual es una técnica mucho más adecuada que la de utilizar solamente una función que agrega un delay a el microcontrolador. Sin embargo por una mala planificación este objetivo no logró ser implementado del todo, a pesar de que se comenzó a trabajar en él, por motivos de funcionamiento, se decide entregar una versión en la que estos timers no son utilizados. Se recomienda utilizar estos timers para poder implementar tanto una mejor temporización de los leds, como agregar la característica que le hizo falta al juego, que es la de reducir el tiempo en 200ms cada ronda.

Referencias

- [1] SN.SF.Microchip ATtiny2313A/4313 Data Sheet 20-Pin 8-bit Microcontroller with 2/4K Bytes In-System Programmable Flash Microcontrollers. Recuperado de:https://mv1.mediacionvirtual.ucr.ac.cr/pluginfile.php/2011142/mod_resource/content/2/ATtiny2313.pdf