

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

II Ciclo 2024

Laboratorio # 2

## GPIOs, Timers y FSM

Jonathan Rodríguez Hernández, B76490 Grupo: 01

Profesor: MSc. Marco Villalta Fallas

14 de abril del 2024

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Nota teórica</b>	<b>2</b>
2.1. Información general MCU . . . . .	2
2.1.1. Diagrama de bloques . . . . .	2
2.1.2. Diagrama de pines . . . . .	3
2.1.3. Características eléctricas . . . . .	4
2.2. Periféricos . . . . .	5
2.3. Registros . . . . .	5
2.4. Diseño de circuito . . . . .	5
2.4.1. Pull-up . . . . .	6
2.4.2. Control de rebotes . . . . .	6
2.4.3. Circuito de LEDs . . . . .	6
2.5. Lista de componentes y precios . . . . .	6
<b>3. Desarrollo/análisis</b>	<b>7</b>
3.1. Análisis programa . . . . .	8
3.2. Análisis electrónico . . . . .	8
<b>4. Conclusiones y recomendaciones</b>	<b>9</b>
<b>5. Bibliografía</b>	<b>10</b>

# 1. Introducción

Se pretende desarrollar un juego de memoria llamado Simon dice, utiliza 4 leds, 4 botones y el microcontrolador ATtiny4313. En este juego hay 4 botones que corresponden a 4 LEDs de colores distintos, la idea consiste en memorizar una secuencia aleatoria de luces para reproducirla posteriormente. Al principio del juego se van encendiendo unos LEDs en un orden y se debe memorizar cuales LEDs se han encendido y en cual orden para reproducir la secuencia. La dificultad va en aumento conforme se va avanzando de modo que al principio se empieza por memorizar cuatro LED y posteriormente se va incrementando el número de LEDs a memorizar. Conforme se va incrementando la secuencia el tiempo que duran los LEDs encendidos se va reduciendo también, al inicio cada LED dura 2 segundos encendidos, posteriormente por cada incremento de la secuencia se debe reducir este tiempo en 200ms (con base a esto se tiene un máximo de 9 niveles de dificultad). Para iniciar el juego se puede presionar cualquier botón y como indicación para el usuario, se parpadean todos los LEDs 2 veces. El juego finaliza cuando el usuario se equivoca en la secuencia, momento en el cual aparece una indicación que consiste en el parpadeo de todos los LEDs 3 veces. La temporización de los LEDs se realiza utilizando los temporizadores del microcontrolador. Para hacer la lectura de los botones, como también del fin de cuenta del timer, se usan interrupciones. Se utiliza un modelo de máquina de estados para la programación del microcontrolador. Durante el desarrollo no se logró utilizar la interrupción PCIE1 de manera efectiva, por lo cual, se optó por hacer una interfaz de 3 botones y 3 LEDs.

Se utiliza el siguiente repositorio: [https://github.com/jodridez/IE-0624\\_IS-2024](https://github.com/jodridez/IE-0624_IS-2024)

## 2. Nota teórica

### 2.1. Información general MCU

El microcontrolador ATtiny4313, basado en la tecnología CMOS, destaca por su eficiencia energética y compacto tamaño con 20 pines de conexión. Es parte de la reconocida familia AVR, y opera con una arquitectura de 8 bits, de manera que posee una memoria flash de 4 KB para programas, 256 bytes de memoria EEPROM para almacenamiento de datos y 256 bytes de SRAM. Cuenta con un oscilador interno, ajustable hasta 20 MHz, al igual que diversas opciones de configuración del sistema de reloj para optimizar el consumo energético y la velocidad de procesamiento según las necesidades del usuario. Incluye funcionalidades como: un módulo USART para comunicaciones seriales. [1]

#### 2.1.1. Diagrama de bloques

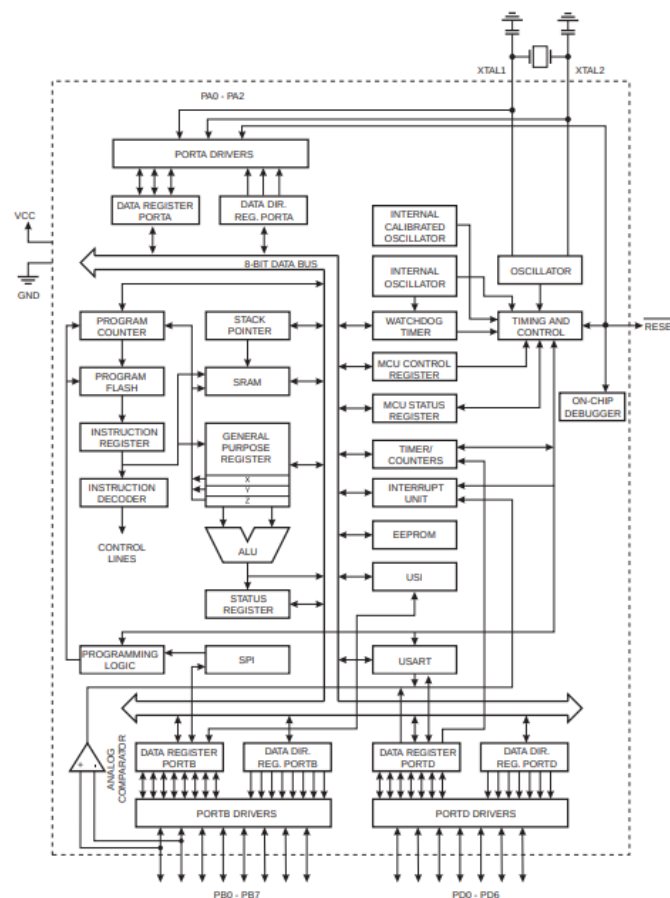


Figura 1: Diagrama de bloques del microcontrolador ATtiny4313

### 2.1.2. Diagrama de pines

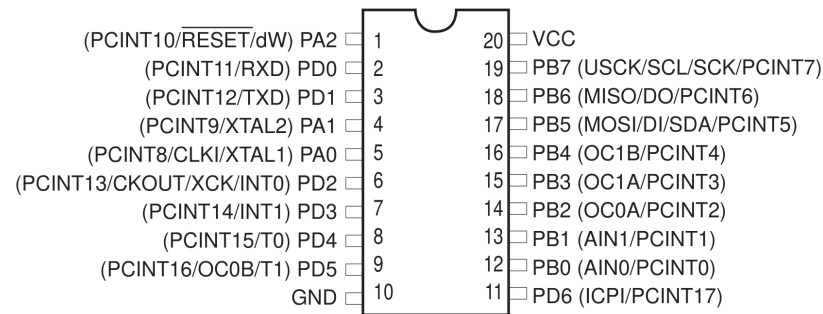


Figura 2: Diagrama de pines del microcontrolador ATtiny4313

### 2.1.3. Características eléctricas

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 3: Valores máximos del ATtiny4313

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
$V_{IL}$	Input Low Voltage except XTAL1 and RESET pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5		$0.2V_{CC}$ $0.3V_{CC}$	V
$V_{IH}$	Input High-voltage except XTAL1 and RESET pins	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(1)}$ $0.6V_{CC}^{(1)}$		$V_{CC} + 0.5^{(2)}$	V
$V_{IL1}$	Input Low Voltage XTAL1 pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.1V_{CC}$	V
$V_{IH1}$	Input High-voltage XTAL1 pin	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.8V_{CC}^{(1)}$ $0.7V_{CC}^{(1)}$		$V_{CC} + 0.5^{(2)}$	V
$V_{IL2}$	Input Low Voltage RESET pin	$V_{CC} = 1.8V - 5.5V$	-0.5		$0.2V_{CC}$	V
$V_{IH2}$	Input High-voltage RESET pin	$V_{CC} = 1.8V - 5.5V$	$0.9V_{CC}^{(1)}$		$V_{CC} + 0.5^{(2)}$	V
$V_{IL3}$	Input Low Voltage RESET pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	-0.5		$0.2V_{CC}$ $0.3V_{CC}$	V
$V_{IH3}$	Input High-voltage RESET pin as I/O	$V_{CC} = 1.8V - 2.4V$ $V_{CC} = 2.4V - 5.5V$	$0.7V_{CC}^{(1)}$ $0.6V_{CC}^{(1)}$		$V_{CC} + 0.5^{(2)}$	V
$V_{OL}$	Output Low Voltage <sup>(3)</sup> (Except Reset Pin) <sup>(5)</sup>	$I_{OL} = 20 \text{ mA}$ , $V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}$ , $V_{CC} = 3V$			0.8 0.6	V V
$V_{OH}$	Output High-voltage <sup>(4)</sup> (Except Reset Pin) <sup>(5)</sup>	$I_{OH} = -20 \text{ mA}$ , $V_{CC} = 5V$ $I_{OH} = -10 \text{ mA}$ , $V_{CC} = 3V$	4.2 2.4			V V
$I_{IL}$	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin low (absolute value)			$1^{(6)}$	$\mu A$
$I_{IH}$	Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin high (absolute value)			$1^{(6)}$	$\mu A$
$R_{RST}$	Reset Pull-up Resistor		30		60	k $\Omega$
$R_{PU}$	I/O Pin Pull-up Resistor		20		50	k $\Omega$
$I_{CC}$	Power Supply Current	Active 1MHz, $V_{CC} = 2V^{(7)}$		0.2	0.55	mA
		Active 4MHz, $V_{CC} = 3V^{(7)}$		1.3	2.5	mA
		Active 8MHz, $V_{CC} = 5V^{(7)}$		3.9	7	mA
		Idle 1MHz, $V_{CC} = 2V^{(7)}$		0.03	0.15	mA
		Idle 4MHz, $V_{CC} = 3V^{(7)}$		0.25	0.6	mA
		Idle 8MHz, $V_{CC} = 5V^{(7)}$		1	2	mA
	Power-down mode	WDT enabled, $V_{CC} = 3V^{(8)}$		4	10	$\mu A$
		WDT disabled, $V_{CC} = 3V^{(8)}$		< 0.15	2	$\mu A$

Figura 4: Características en DC del ATtiny4313

## 2.2. Periféricos

- Temporizadores/Contadores.
- Sistema de interrupciones.
- Puertos de I/O.

## 2.3. Registros

**DDRx:** El Registro de Dirección de Datos, define si los pines son entradas o salidas.

**PORTx:** El Registro de Datos del Puerto, controla los valores de salida o activa las resistencias de pull-up para pines de entrada.

**sei():** Habilita las interrupciones globales.

**GIMSK:** El Registro de Máscara de Interrupción General permite escoger cuales interrupciones se habilitaran.

**MCUCR:** El Registro de Control del MCU gestiona la configuración de las interrupciones correspondientes a INT0 o INT1, por ejemplo, permite escoger si la interrupción se activa por flanco de subida.

**TCCR0A:** El Registro de Control del Temporizador/Contador 0A, configura el temporizador/-contador 0. Se configuró en modo CTC. temporizador incrementa su cuenta.

**TCCR0B:** Registro para configurar la fuente de reloj. Se configuro para utilizar el reloj del sistema con un prescaler de 1024.

**TIMSK:** El Registro de Máscara de Interrupción del Temporizador/Contador controla las interrupciones relacionadas con los temporizadores/contadores. Se configuró para habilitar la interrupción por comparación.

**OCR0A:** El Registro de Comparación de Salida 0A define el valor de comparación para el temporizador/contador 0.

## 2.4. Diseño de circuito

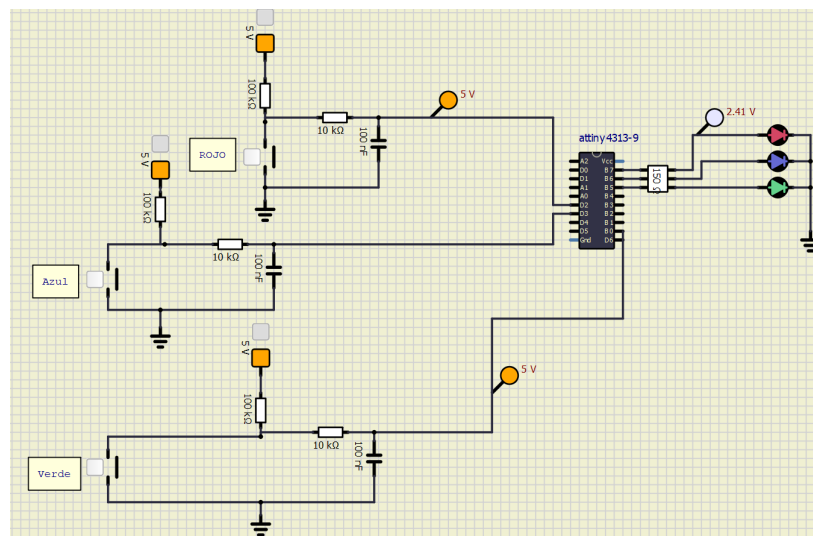


Figura 5: Esquemático del circuito diseñado

### 2.4.1. Pull-up

Se utilizó una configuración de entrada pull-up, el estado esta en alto hasta que ocurre un evento que lo pone en bajo, en este caso, hasta que se presiona algún botón.

### 2.4.2. Control de rebotes

Para el control de rebotes se incluyo un capacitor y se siguió las recomendaciones de la guía: [2].

### 2.4.3. Circuito de LEDs

Se utilizaron 3 LEDs (rojo, azul, verde). Cada pin de salida actúa como una fuente de tensión de 5 V con corriente máxima de 40 mA (ver características eléctricas). Se calcula la resistencia de protección necesaria:

$$R = \frac{5 - V_{LED}}{I} \quad (1)$$

$$R = \frac{5 - (2)}{20 \times 10^{-3}} = 150 \quad \Omega \quad (2)$$

## 2.5. Lista de componentes y precios

Componente	Cantidad	Precio Individual (USD)
ATtiny4313	1	1.80
LED	3	0.20
Botón	3	0.10
Resistencia 10 k $\Omega$	3	0.06
Resistencia 100 k $\Omega$	3	0.06
Resistencia de 150 $\Omega$	3	0.06
Total aproximado		3.9

Tabla 1: Lista de componentes y precios estimados



### 3. Desarrollo/análisis

El programa desarrolla el juego Simon dice utilizando un microcontrolador tiny4313. Se estructura en una máquina de estados:

1. WAITING: Estado inicial, se hace reset de las variables y se espera a que el usuario presione algún botón para pasar al estado INIT.
2. INIT: Los LEDs parpadean 2 veces indicando el inicio del juego. Se pasa al estado GENERATE\_SEQUENCE
3. GENERATE\_SEQUENCE: Se generan números aleatorios del 1 al 3, representando los LEDs, y se introducen en un vector. Se pasa al estado SHOW\_SEQUENCE.
4. SHOW\_SEQUENCE: Cada LED parpadea siguiente la secuencia generada, se tiene duración determina según el nivel de dificultad, con un máximo de 200 ms de duración. Se pasa al estado USER\_INPUT.
5. USER\_INPUT: El usuario debe presionar los botones que corresponden a los LEDs, cada entrada es guardada en un vector que se utilizara para comparar con el vector generado por el juego anteriormente. Cada vez que se pulsa un botón se ilumina el LED correspondiente con el fin de darle retroalimentación al jugador. Una vez introducida la secuencia, se pasa al estado CHECK\_SEQUENCE.
6. CHECK\_SEQUENCE: Se comparan ambos vectores, en caso de haber concordancia, se sube el nivel de dificultad y se pasa al estado GENERATE\_SEQUENCE, en caso contrario, se pasa al estado GAME\_OVER.
7. GAME\_OVER: Parpadean todos los LEDs 3 veces y se pasa al estado WAITING.

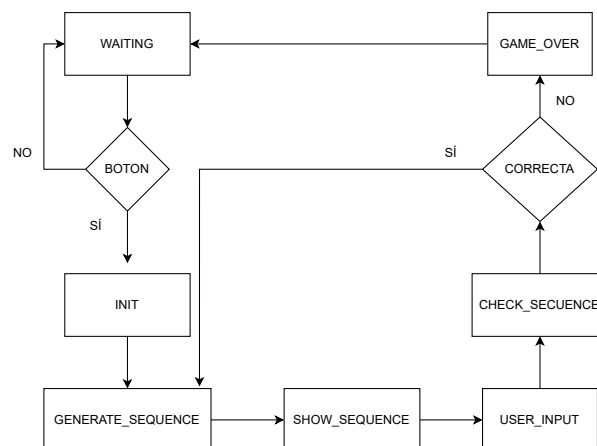


Figura 6: Diagrama de flujo de la FMS

### 3.1. Análisis programa

El programa cuenta con errores, los cuales se anotan a continuación:

1. No se logra hacer uso de la interrupción PCIE1, el vector de interrupción del PCINT8 no era identificado al copilar, se probaron varios PCINTx pero solo los vectores de PCINT0 y PCINT1 se lograron usar, motivo por el cual se terminó optando por realizar el juego con solo 3 colores ya que PCINT0 y PCINT1 disparaban la misma interrupción.
2. Al pasar al estado SHOW\_SEQUENCE por primera vez siempre se muestra la misma secuencia, 3 rojos, al presionar el rojo una sola vez, se mueve hasta al estado SHOW\_SEQUENCE de nuevo, a partir de ahí el programa funciona como se debe hasta que se falle por primera vez.
3. Al fallar por primera vez, se recorren los estados con normalidad hasta llegar al estado USER\_INPUT, sin embargo, al presionar cualquier botón una sola vez el programa se mueve al estado GAME\_OVER, haciendo que el juego realmente solo se pueda reiniciar al reiniciar toda la simulación.

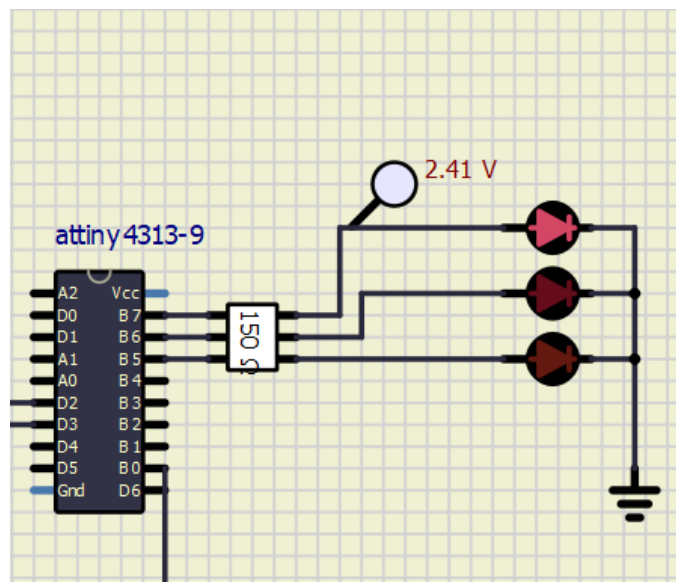


Figura 7: LED rojo parpadeando 3 veces

### 3.2. Análisis electrónico

Se muestra en la figura 5 que las tensiones no superan los máximos absolutos.

## 4. Conclusiones y recomendaciones

Se desarrolla un Simón dice con una interfaz de 3 colores, y con errores anteriormente mencionados. se logran cumplir los siguientes puntos.

- Utilizar interruptores para leer el estado de los botones, así como la fin de cuenta de los timers.
- Programación de una FSM utilizando la estructura de control switch.

## 5. Bibliografía

### Referencias

- [1] ATtiny4313 Datasheet, Atmel Corporation, 2007. [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/doc8246.pdf>
- [2] J. Christoffersen, “Switch bounce: How to deal with it,” All About Circuits, s/f.