

**EIE**

Escuela de  
Ingeniería Eléctrica

**Universidad de Costa Rica**

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica



UNIVERSIDAD DE  
COSTA RICA

**IE-0624**

Laboratorio de Microcontroladores

## **Laboratorio #2**

Fernando Jiménez Ureña B74020

Kristel Herrera Rodríguez C13769

II ciclo

Septiembre 2023

# 1. Resumen

En el presente laboratorio se desarrolló mediante el microcontrolador AT-tiny4313 un cruce de semáforos simplificado. El objetivo principal del laboratorio fue el estudio más a profundidad de temas como los GPIOs de un microcontrolador que se empezó a estudiar en el laboratorio 1, y la introducción de nuevos temas como fue el caso de interrupciones, timers y máquinas de estado en un microcontrolador.

Se utilizó el **software de simulación SimulIDE** para el diseño del circuito y se utilizaron elementos como luces LED, botones, capacitores y resistencias además del microcontrolador para poder construir existosamente el diseño. Además se utilizó el **lenguaje de programación C** para el desarrollo del código y poder programar las funcionalidades de los diferentes semáforos.

En resumen, el funcionamiento consiste en que se tienen dos semáforos peatonales y un semáforo vehicular. Al inicio de la simulación, se debe tomar que por al menos 10 segundos el semáforo vehicular debe estar en verde. Al pasar estos 10 segundos, se procede a presionar cualquiera de los dos botones de los semáforos peatonales indicando una interrupción y se empieza a correr la lógica implementada para el funcionamiento de los semáforos peatonales que será descrita con más profundidad más adelante en este reporte.

Se puede encontrar el link del repositorio donde se encuentra contenido el proyecto en: <https://github.com/KrisKy02/Laboratorio2-microcontroladores>

## 2. Nota Teórica

### 2.1. Características Generales

El microcontrolador ATtiny4313 de Atmel es una unidad de 8 bits diseñada para ofrecer un alto rendimiento. Gracias a su arquitectura RISC avanzada, el dispositivo ofrece una programación en sistema altamente flexible. Además, cuenta con una amplia gama de periféricos incorporados y es notable por sus múltiples opciones para el bajo consumo de energía. Todas las características técnicas descritas en esta sección se basan en la hoja de datos oficial del microcontrolador ATtiny4313 [1].

### 2.2. Arquitectura RISC Avanzada

- 120 instrucciones potentes, la mayoría con ejecución de un solo ciclo de reloj
- 32 x 8 registros de trabajo de propósito general
- Operación completamente estática
- Hasta 20 MIPS de rendimiento a 20 MHz

### 2.3. Memorias de Programa y Datos No Volátiles

- 2/4K Bytes de Flash programable en el sistema
- Durabilidad de 10,000 ciclos de escritura/borrado
- 128/256 Bytes de EEPROM programable en el sistema
- Durabilidad: 100,000 ciclos de escritura/borrado
- 128/256 Bytes de SRAM interna

### 2.4. Características Periféricas

- Un contador de temporizador de 8 bits con preescalador y modo de comparación separados
- Un contador de temporizador de 16 bits con preescalador, modos de comparación y captura separados
- Cuatro canales PWM
- Comparador analógico en chip
- Temporizador de vigilancia programable con oscilador en chip
- USI: Interfaz Serial Universal
- USART dúplex completo

## 2.5. Características Especiales del Microcontrolador

- debugWIRE para depuración en chip
- Programable en el sistema a través del puerto SPI
- Fuentes de interrupción externas e internas
- Modos de bajo consumo: inactivo, apagado y en espera

## 2.6. I/O, Voltaje de Operación y Otras Especificaciones

- 18 líneas I/O programables
- Paquetes de 20 pines PDIP, SOIC, y 20-pads MLF/VQFN
- Voltaje de operación entre 1.8 – 5.5V
- Grados de velocidad:
  - 0 – 4 MHz a 1.8 – 5.5V
  - 0 – 10 MHz a 2.7 – 5.5V
  - 0 – 20 MHz a 4.5 – 5.5V
- Rango de temperatura industrial: -40°C a +85°C
- Bajo consumo de energía:
  - Modo Activo: 190  $\mu$ A a 1.8V y 1MHz
  - Modo Inactivo: 24  $\mu$ A a 1.8V y 1MHz
  - Modo de Apagado: 0.1  $\mu$ A a 1.8V y +25°C

## 2.7. Diagrama de Pines

A continuación se presenta el diagrama de pines del ATtiny4313:

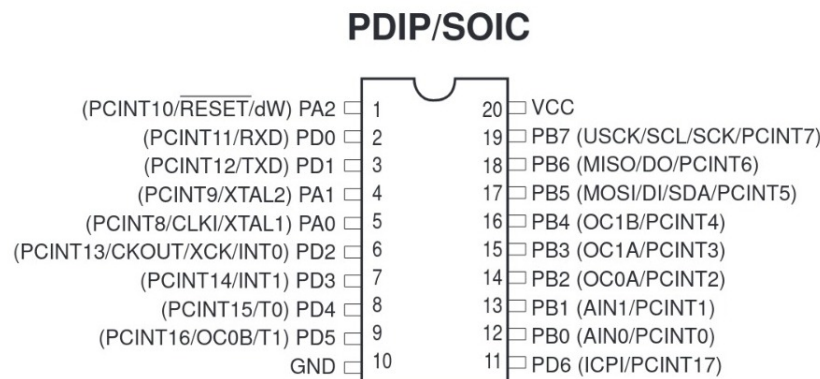


Figura 1: Diagrama de pines[1].

Para el presente laboratorio, se utilizaron los siguientes puertos: PB0, PB1, PB2, PB3, y PD2.

## Port B (PB7..PB0)

Port B es un puerto de E/S bidireccional de 8 bits con resistencias pull-up internas que pueden ser seleccionadas para cada bit. Los búferes de salida de Port B presentan características de manejo simétricas, con alta capacidad tanto para hundir como para suministrar corriente. Cuando se utilizan como entradas, los pines de Port B que están externamente tirados a bajo generarán corriente si las resistencias pull-up están activadas. Los pines de Port B entran en estado de alta impedancia cuando se activa una condición de reinicio, incluso si el reloj no está en funcionamiento [1].

## Port D (PD6..PD0)

Similar a Port B, Port D es un puerto de E/S bidireccional de 7 bits con resistencias pull-up internas seleccionables para cada bit. Los búferes de salida de Port D también tienen características simétricas de manejo de corriente. Al igual que con Port B, los pines de Port D que están externamente tirados a bajo generarán corriente si se activan las resistencias pull-up. Los pines de Port D también se ponen en estado de alta impedancia durante una condición de reinicio, incluso si el reloj no está funcionando [1].

### 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	

### 10.3.6 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	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	

### 10.3.7 PINB – Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

### 10.3.8 PORTD – Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x12 (0x32)	–	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### 10.3.9 DDRD – Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x11 (0x31)	–	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

### 10.3.10 PIND – Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x10 (0x30)	–	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Figura 2: Descripción de pines[1].

## 2.8. Diagrama de bloques

A continuación se presenta el diagrama de bloques del ATtiny4313:

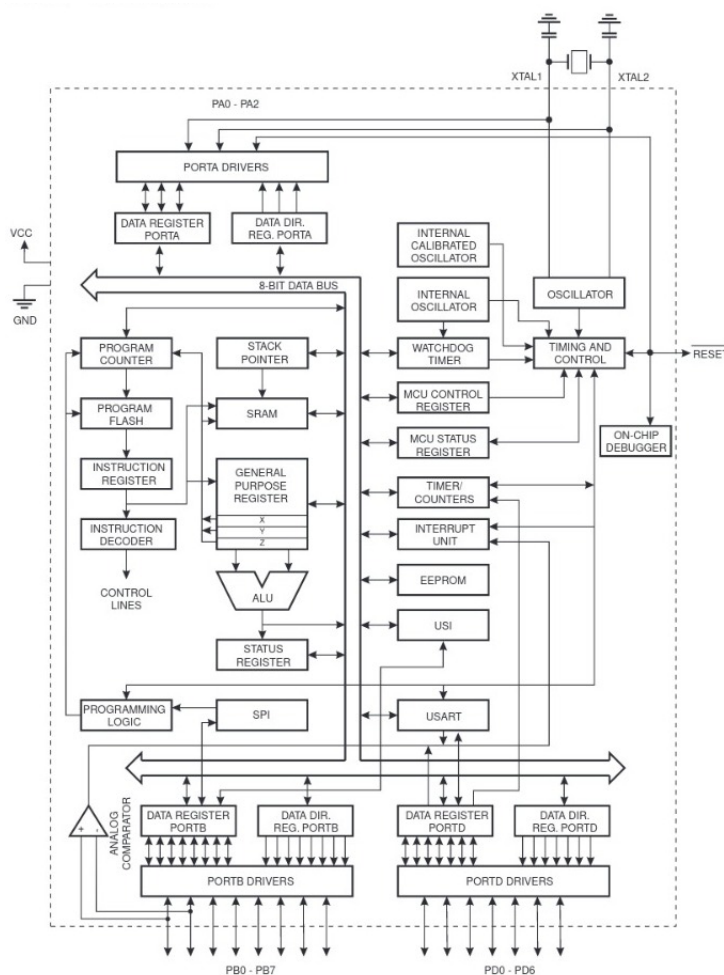


Figura 3: Diagrama de bloques [1].

## 2.9. Interrupciones

Las interrupciones son fundamentales para el funcionamiento eficiente de un microcontrolador. Cuando se activa una interrupción, la ejecución normal del programa se detiene temporalmente. El estado actual del programa, incluidas las variables y el contador de programa, se guarda en un bloque de memoria especial. Luego, el microcontrolador busca en este bloque para encontrar la Rutina de Servicio de Interrupción (ISR) correspondiente que necesita ser ejecutada [2].

Existen cuatro componentes clave en el mecanismo de interrupción:

1. **Banderas de Interrupción:** Indican si ha ocurrido un evento de interrupción.

2. **Bits de Habilidad de Interrupción:** Determinan si una interrupción específica será atendida automáticamente por el hardware.
3. **Habilitación Global de Interrupciones:** Permite o impide que ocurran interrupciones en el sistema.
4. **Rutina de Servicio de Interrupción (ISR):** Es el conjunto de instrucciones que se ejecuta cuando se produce una interrupción [2].

La ejecución eficiente de las ISRs es crucial para el rendimiento del sistema, ya que evita pérdida de datos o comportamientos inesperados. Estas rutinas son responsables de guardar y restaurar el contexto del programa, permitiendo que el microcontrolador continúe su ejecución donde la dejó [2].

## 2.10. Interrupciones Externas

Las interrupciones externas son mecanismos que permiten a un microcontrolador detectar un cambio de estado en alguna de sus terminales de entrada, evitando así la necesidad de un sondeo continuo. Estas son especialmente útiles para monitorear estados de interruptores, botones, o sensores con salida a relevador. [3]

## 2.11. Configuración de las interrupciones

Las interrupciones externas pueden configurarse para detectar un nivel bajo de voltaje o una transición en el flanco de subida o bajada. Sin embargo, la interrupción INT2 sólo puede activarse por flancos. Importante resaltar que estas interrupciones pueden generarse incluso cuando sus terminales respectivas son configuradas como salidas.

Las transiciones en INT0/INT1 requieren de una señal de reloj (clkI/O) para generar una interrupción. Esta señal de reloj es anulada en la mayoría de los modos de bajo consumo. Por otro lado, un nivel bajo en INT0/INT1 y las transiciones en INT2 no requieren de una señal de reloj, siendo eventos asíncronos y adecuados para despertar al microcontrolador en cualquier modo de reposo.[3]

10.3.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
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: Configuración del MCUCR[1].

9.3.2 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: Configuración del GIMSK[1].

**Table 9-3.** Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Figura 6: Configuración del INT0[1].

### 2.11.1. Configuraciones Específicas

**NT0: External Interrupt Request 0 Enable** Cuando este bit y el bit I en el Registro de Estado (SREG) están activados, se habilita la interrupción externa en el pin INT0. Los bits de Control de Sentido de Interrupción (ISC01 y ISC00) en el Registro de Control de Interrupción Externa A (EICRA) determinan si la interrupción se activa en los bordes ascendentes y/o descendentes del pin INT0.

**PUD: Pull-up Disable** Al escribir un “uno” en este bit, se desactivan las resistencias *pull-up* en los puertos de E/S.

**ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0** La Interrupción Externa 0 se activa por el pin externo INT0 si la bandera I del SREG y la máscara de interrupción correspondiente están establecidas.

**Duración de la Señal para Interrupción** Si se selecciona una interrupción de borde o de conmutación, los pulsos que duran más de un período de reloj generarán una interrupción. Los pulsos más cortos no garantizan la generación de una interrupción.

## 2.12. Temporización en Microcontroladores ATMega

### 2.12.1. Importancia de la Temporización Precisa

Mantener un tiempo preciso es crucial, especialmente en tareas que requieren mediciones exactas o en sistemas de navegación. Los temporizadores internos del microcontrolador permiten una gestión del tiempo altamente precisa, esencial para el control eficaz del sistema [2].

### 2.12.2. Tipos de Temporizadores

**Temporizadores Internos** Los microcontroladores ATMega8 y ATMega16 cuentan con temporizadores 0 y 1 que pueden funcionar como contadores de eventos externos.

**Temporizadores Externos** Están diseñados para funcionar con señales externas a través de los pines T0 y T1. Además, el temporizador 2 puede operar con un oscilador externo de 32.768 KHz.



### 2.12.3. Modos de Operación

- **Modo 0:** Operación básica con eventos de desbordamiento.
- **Modos 1 y 3:** Utilizados para la generación de señales PWM.
- **Modo 2:** Permite limpiar el registro del temporizador en coincidencias específicas [3].

### 2.12.4. Registros Importantes

El temporizador 1 utiliza registros de 16 bits como TCNT1, OCR1A, OCR1B e ICR1 que son esenciales para su funcionamiento. Los registros TCCR1A y TCCR1B controlan el comportamiento del temporizador.

#### 11.9.3 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 7: Configuración del TCNT0[1].

#### 11.9.1 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 8: Configuración del TCCR0A[1].

**Table 11-2.** Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Figura 9: Configuración del COMA0[1].

**11.9.2 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 10: Configuración del TCCR0B[1].

**Table 11-5.** Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

Figura 11: Configuración del COMB0[1].

**2.12.5. Selección de la Fuente del Reloj**

Para el temporizador 0, la selección del reloj se controla con los bits CS0[2:0], y para el temporizador 1, con los bits CS1[2:0].

### 3. Desarrollo

#### 3.1. Diseño

Con el diseño implementado los dos botones al presionarse van a conectar el pin D2 a +5V, y la corriente máxima que estos pines pueden proporcionar es de 40 mA. Sin embargo para asegurar el funcionamiento se limitó la corriente a 30 mA.

El valor de la tensión de los LEDs es de 2.0 V. Por lo tanto, utilizando Ley de Kirchhoff para calcular el valor de las resistencias para el caso de un solo LED:

$$R = \frac{V_{DD} - V_{LED}}{I_{Max}} = \frac{5 - 2,0}{25mA} = 120\Omega \quad (1)$$

Tomando los valores en bodega de la escuela, se eligieron entonces resistencias de para cada uno de los LEDs de  $120\Omega$ . Estas resistencias son de gran importancia para proteger los LEDs.

Además, es de suma importancia notar que un circuito RC se agregó para proteger de los efectos rebote a los botones. Se definió una resistencia de  $10K\Omega$  debido a que proporciona una corriente limitada a través del circuito para reducir el consumo de energía y la carga en la fuente de alimentación, seguidamente el capacitor de 10 nF ofrece un buen equilibrio entre filtrado y velocidad de respuesta de los botones. [4]

El tiempo que transcurre desde que se presiona el pulsador momentáneo hasta que el condensador se carga por completo se puede calcular utilizando la siguiente fórmula:

$$Tiempo(segundos) = Valor de R \cdot Capacidad del condensador \quad (2)$$

Calculando con los valores que se diseñaron para el circuito:

$$Tiempo(segundos) = 10K\Omega \cdot 10nF \quad (3)$$

$$Tiempo(segundos) = 10K\Omega \cdot 10nF \quad (4)$$

$$Tiempo(segundos) = 0,1s \quad (5)$$

Este tiempo permite una respuesta lo suficientemente rápida para que el semáforo peatonal se active después de que el usuario presione el botón, y al mismo tiempo elimina de manera efectiva cualquier rebote en el interruptor

A nivel del código implementado, se puede mencionar que se agregaron ciertas características para poder optimizar el rendimiento y código. Entre ellas destacan el uso de macros para definir valores constantes y representar unidades de tiempo clave.

El uso de macros permiten asignar nombres significativos a valores numéricos, lo que hace que el código sea más legible, reutilizable y fácil de mantener. Además, las macros facilitan la modificación de estos valores en un solo lugar, lo que simplifica la adaptación del comportamiento del semáforo sin necesidad de buscar y cambiar cada instancia numérica en el código. Esta característica puede mejorar la escalabilidad del código, ya que cualquier ajuste necesario en los tiempos de cambio de estado o ciclos del temporizador se puede realizar de manera eficiente y precisa simplemente modificando las macros.

Además otra implementación importante fue el uso de funciones *inline* para funcionalidades como el parpadeo de las luces del semáforo y la gestión de las interrupciones. Este tipo de funciones se utilizan para evitar la sobrecarga de llamadas a funciones y, en su lugar, insertar directamente el código de la función en el lugar donde se invoca. Esto ahorra el tiempo y los recursos que normalmente se consumirían en la llamada y retorno de una función, lo que resulta en un código más rápido y eficiente en términos de rendimiento. [5]

A continuación se muestran los costos de los componentes utilizados:

Componente	Precio
ATtiny4313	583 colones
Resistencia de 120 Ohms	155 colones
Botón	150 colones
Resistencia de 10 KOhms	120 colones
10 nF capacitor	325 colones
LED Verde	310 colones
LED Rojo	310 colones

Tabla 1: Precio de los componentes

### 3.2. Funcionalidad del diseño

El circuito en el simulador se muestra de la siguiente manera:

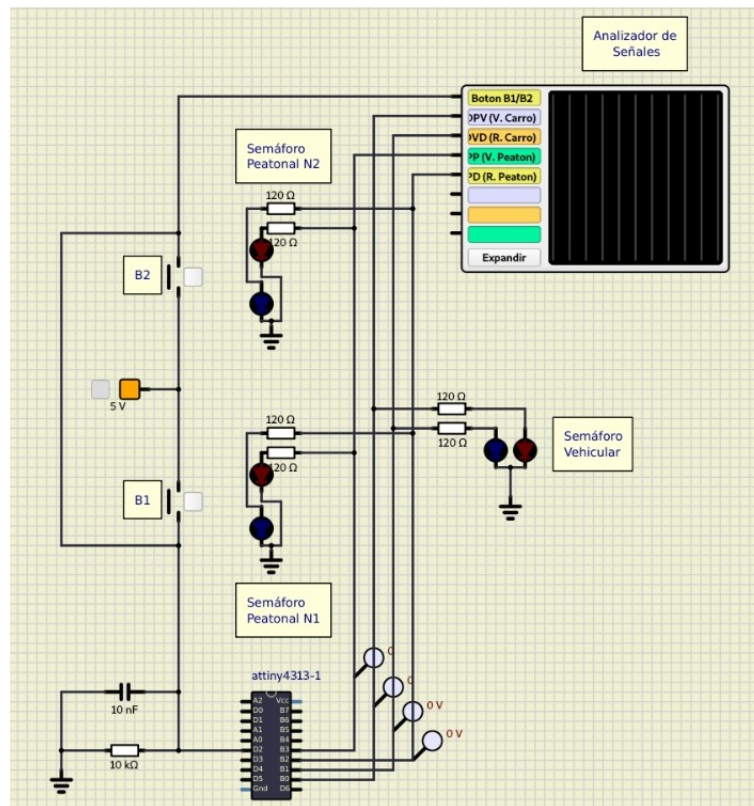


Figura 12: Diseño del circuito.

Al poner a funcionar el simulador, se puede observar el estado inicial del semáforo a continuación. Puede notarse que en su estado inicial los semáforos peatonales se encuentran en rojo mientras que el semáforo vehicular en verde

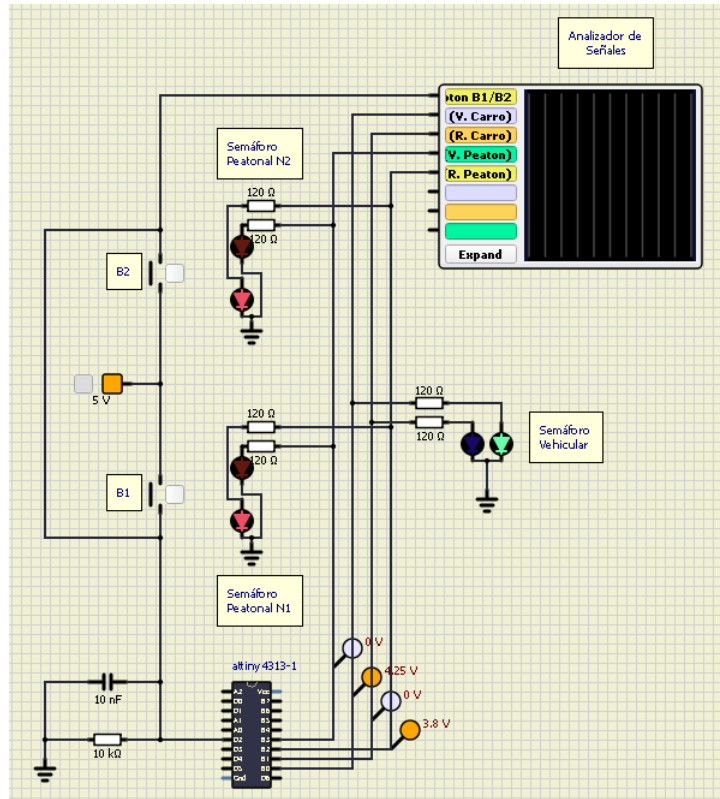


Figura 13: Estado inicial del semáforo

Seguidamente, al presionar cualquiera de los dos botones se puede observar el semáforo peatonal en funcionamiento. El semáforo vehicular pasa a rojo mientras que los peatonales a verde

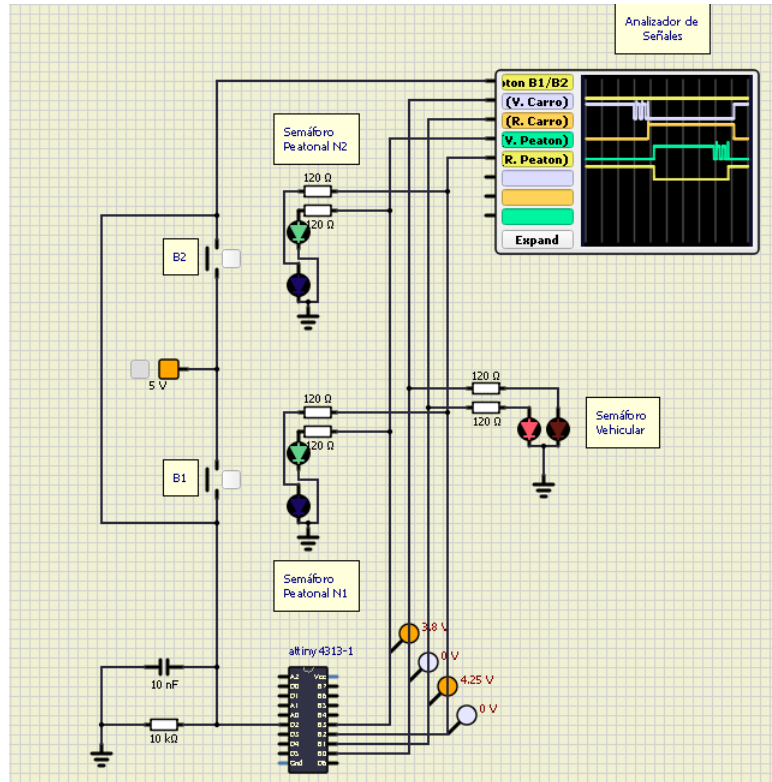


Figura 14: Semáforos en funcionamiento

Finalmente, luego del tiempo definido para el funcionamiento de los semáforos peatonales, éstos vuelven a su estado inicial y el semáforo vehicular vuelve a estar en verde.

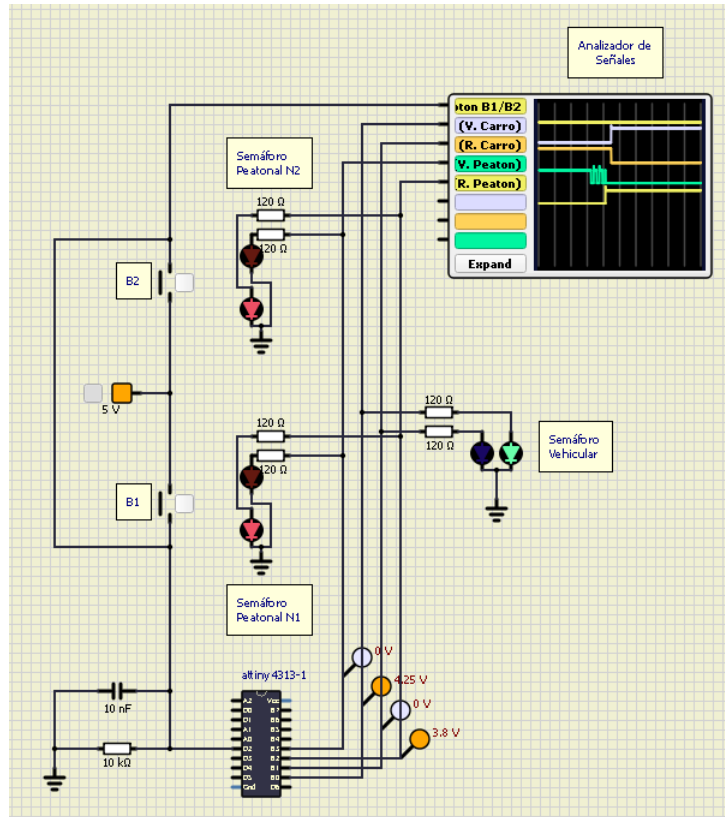


Figura 15: Semáforos en funcionamiento



### 3.3. Diagrama de estados

A continuación se presenta el diagrama de estados del proyecto:

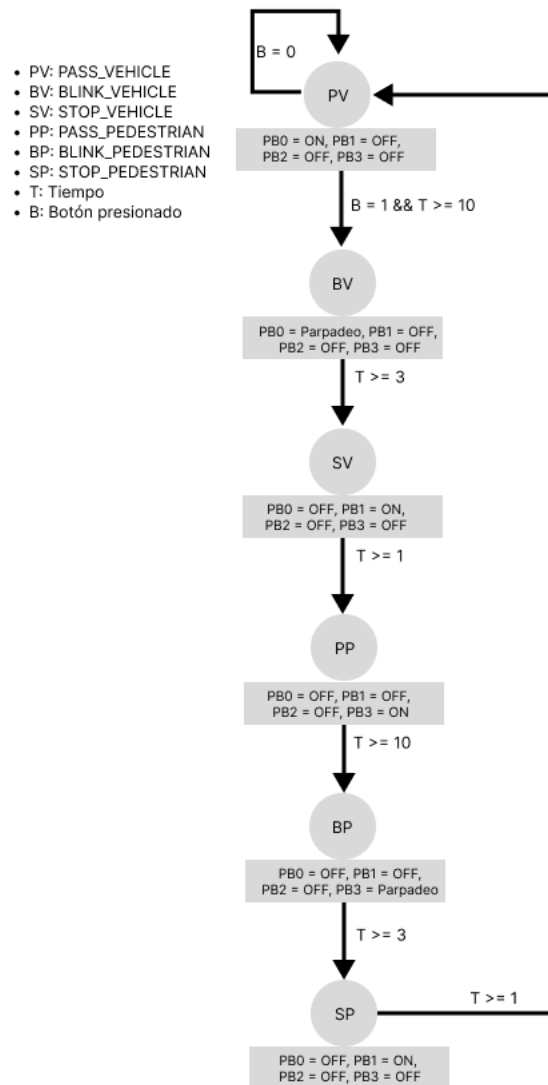


Figura 16: Diagrama de estados.

### 3.4. Funcionalidad electrónica

Se colocó un osciloscopio en las salidas de cada pin y en la entrada del botón para mostrar que los cambios en la tensión son los esperados. En la siguiente imagen se puede observar la lectura del osciloscopio.

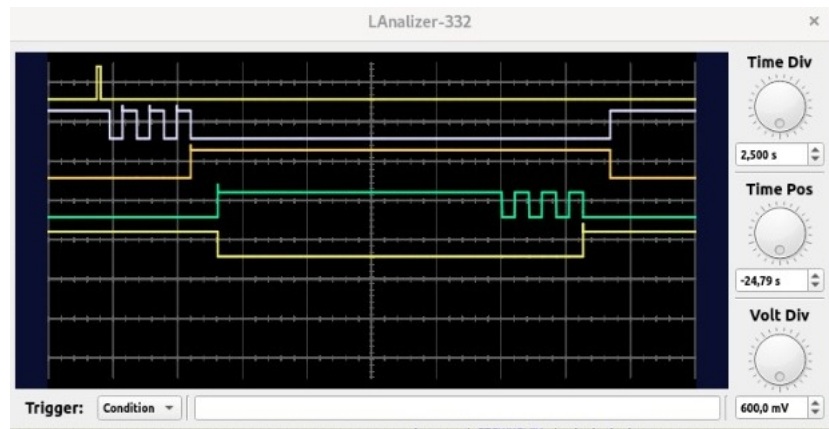


Figura 17: Lectura del osciloscopio.

Al inicio de las pruebas, todas las salidas se encontraban en un estado lógico bajo ('0'), cumpliendo con las expectativas iniciales. Del mismo modo, el botón de entrada también comenzó en un estado bajo. Cuando se presionó el botón, la señal asociada mostró un flanco ascendente, que activó una transición en la máquina de estados. Durante el experimento, las transiciones entre diferentes estados de las salidas se llevaron a cabo de manera coherente con la definición de la máquina de estados.

Además, se pudo verificar que cada estado se mantenía durante el período de tiempo especificado antes de avanzar al siguiente estado. Es importante destacar que durante las pruebas no se detectaron falsas transiciones ni ruido eléctrico que pudieran afectar la secuencia de estados. Esto indica que el sistema de "debouncing" para el botón funciona correctamente, y que la implementación de la máquina de estados funciona de la manera esperada.

## 4. Conclusiones y Recomendaciones

- El diseño del circuito demostró ser eficaz y económico, optimizando la corriente de los LEDs y minimizando el ruido y los efectos de rebote en los botones. La selección de resistencias y el uso de un circuito RC contribuyeron al rendimiento óptimo del sistema.
- La funcionalidad de la máquina de estados se validó a través de pruebas con un osciloscopio, confirmándose que las transiciones de estado y la duración de los estados fueron las esperadas. No se observaron ruidos ni falsas transiciones.
- Se logró cumplir de manera correcta el diagrama de temporización propuesto en el enunciado del laboratorio
- Se logró construir un diseño funcional para el microcontrolador ATtiny4313, logrando así conocer sus funcionalidades y características.

## Referencias

- [1] A. Corporation, *ATtiny4313 Datasheet*, Atmel Corporation, 2011, [Consultado el: 1 de septiembre del 2023].
- [2] P. Alley, “Introductory microcontroller programming,” Master’s Thesis, Worcester Polytechnic Institute, May 2011.
- [3] F. Santiago. (Año de Publicación) Libro sobre Controles Digitales: Capítulos 4, 5, 6, 7. Accedido el día [Fecha de acceso]. [Online]. Available: [http://controlesdigitales.com/Libro\\_Felipe\\_Santiago/04\\_Cap\\_4\\_5\\_6\\_7.pdf](http://controlesdigitales.com/Libro_Felipe_Santiago/04_Cap_4_5_6_7.pdf)
- [4] D. Maker. (2014) Debouncing de un pulsador momentáneo vía hardware. [Online]. Available: <https://drmaker.wordpress.com/2014/03/29/debouncing-de-un-pulsador-momentaneo-via-hardware/>
- [5] Microsoft. (Sin fecha) Funciones en línea (inline) en c++. [Online]. Available: <https://learn.microsoft.com/es-es/cpp/cpp/inline-functions-cpp?view=msvc-170>

## 5. Apéndices

Figura 18: Descripción arquitectónica. Creación propia.