Universidad de Costa Rica

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

Laboratorio 2: GPIOs, Timers y FSM

Gokeh Ávila Blanco, B50747

Profesor: MSc. Marco Villalta Fallas

11 de Mayo de 2022

Índice

Índice de figuras

1. Introducción

En este laboratorio se implementa un semáforo peatonal que esta constituido de un semáforo para los vehículos con los colores verde, amarillo y rojo, y dos semáforos para los peatones con los colores verde y rojo. Para esta implementación se utilizó el microcontrolador ATtiny4313. Como el semáforo que se está implementando es un semáforo peatonal, tenemos que este cuenta con un botón que permite a los peatones solicitar el paso. Por otro lado tenemos que para manejar los tiempos de las luces de los semáforos y para la lectura del botón se utilizan las interrupciones del microcontrolador.

2. Nota teórica

Tenemos que el ATtiny4313 es un microprocesador de 18 pines y 8 bits con lógica CMOS. Por otro lado, viendo la hoja del fabricante [1], este microcontrolador es capaz de alcanzar velocidades de 1 MIPS por MHz, lo cual permite optimizar el circuito para consumo de potencia sobre velocidad de procesamiento.

Algunas características generales importantes del microcontrolador ATtiny4313:

Tensión de alimentación	1.8 - 5.5 V
Frecuencia máxima de operación	20 MHz
Memoria de programa	2/4 kB
Memoria de datos	EEPROM: 128/256 B - RAM: 128/256 B
Pines I/O	18
Temperatura de operación	-40°C a +85°C

2.1. Periféricos utilizados

2.1.1. GPIO del puerto B

Luego tenemos que para el desarrollo del laboratorio se utilizaron los pines I/O del puerto B para controlar las luces del semáforo, para lo cual se utilizaron los registros que se muestran a continuación:

DDRB: Este registro nos permite definir si los pines del puerto B van a ser de entrada o de salida.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0

PORTB: Con este registro se puede leer y escribir el valor en cada uno de los pines del puerto B.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0

2.1.2. Interrupción externa

Para efectos de las interrupciones externas tenemos que se usaron ambas interrupciones externas que tiene el microprocesador para implementar los botones de solicitud de paso para los peatones. Dichas interrupciones están en los pines 3 y 2 del puerto D, y para usar estas interrupciones se usaron los siguientes registros:

GIMSK: Registro que permite habilitar las interrupciones externas por medio de los bits INT0 e INT1.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INT1	INT0	PCIE0	PCIE2	PCIE1	-	-	-

MCUCR: Con este registro se selecciona el flanco con el cual se activa la interrupción externa através de los bits: ISC11, ISC10, ISC01 e ISC00.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00

2.1.3. Timer/Counter Output Compare

Luego, con respecto a los tiempos de espera de los semáforos se usó una interrupción de output compare, para la cual se usaron los siguientes registros:

OCR0A: Registro que permite especificar el valor al cual debe llegar el contador TCNT para disparar la interrupción.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OCR0A[7:0]							

Por otro lado, tenemos que se buscó generar una interrupción de $T_{INT} = 10ms$, y como se usó un reloj $clk_{I/O} = 8MHz$ y el registro TCNT se configuró para que aumentara cada $\frac{clk_{I/O}}{1024}$, entonces tenemos que:

$$T_INT = \frac{1024}{8MHz}.OCR0A$$
$$OCR0A = \frac{8MHz}{1024}.T_INT$$
$$OCR0A = \frac{8MHz}{1024},10ms$$
$$OCR0A = 78$$

TIMSK: Registro que permite activar la interrupción de output compare 0A mediante el bit OCIE0A.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TOIE1	OCIE1A	OCIE1B	-	ICIE1	OCIE0B	TOIE0	OCIE0A

TCCR0A: Registro que deja configurar la interrupción de output compare en modo CTC (clear time compare match) para que la interrupción se dispare cuando TCNT sea igual al valor cargado en OCR0A. Para esto se tiene que cargar un 2 en los registros WGM01:WGM00.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00

TCCR0B: Registro que deja escoger la fuente del reloj que va a ser la base para la cuenta del TCNT, esto através de los bits CS02:CS00.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00

2.1.4. Diseño del circuito

En la figura 1 podemos ver el circuito diseñado para este laboratorio.

El circuito cuenta con 7 LEDs para los semáforos, 3 que son para el semáforo vehicular (3 colores: verde, amarillo y rojo) y 4 LEDs para los dos semáforos peatonales (2 luces por semáforo: verde y rojo). Dichos LEDs tiene una resistencia de 100Ω conectada en serie que sirve como reguladora de corriente para evitar que se dañen.

Por otro lado, el circuito cuenta con dos botones conectados a los pines D3 y D2, con resistencias de pull-down, que permiten activar la interrupción externa para la solicitud del paso peatonal. Por otro lado, estos botones están conectados a un circuito RC que ayuda a la cancelación de rebotes, con R = 1kΩ y C = 10μF, se logra conseguir una curva de 0 a 5V de aproximadamente 20ms, lo cual es un tiempo adecuado para manejar los rebotes del botón.

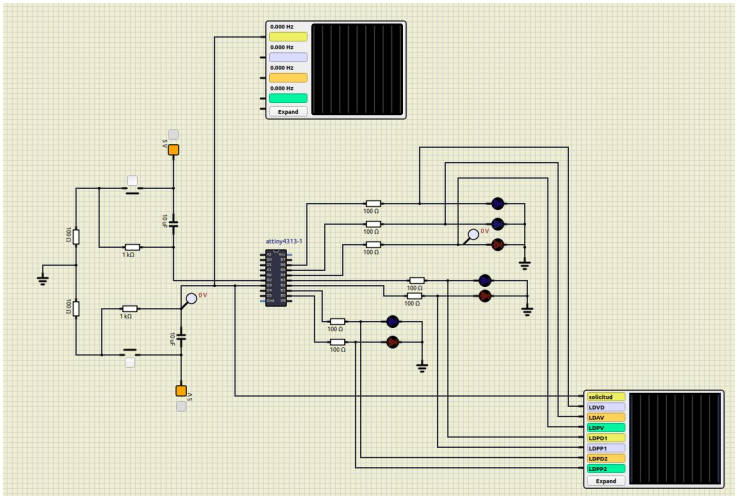

Figura 1: Diseño del semáforo.

2.1.5. Lista de componentes y precios

Los componentes utilizados son los siguientes:

- Resistencias 100Ω - US$0.019 c/u
- Resistencias 1kΩ - US$0,20 c/u
- Capacitores 10μF - US$0,28 c/u
- LEDs verdes - US$0.19 c/u
- LEDs rojos - US$0.19 c/u
- LEDs amarillos - US$0.19
- Botones - US$0.081 c/u
- ATtiny4313 - US$1.411

3. Análisis de resultados

3.1. Máquina de estados

Para implementar el semáforo se utilizó una máquina de estados finitos. Dicha máquina cuenta con 6 estados, los cuales podemos ver en el diagrama de la figura 2. Por otro lado, tenemos que los patrones de luces son los siguientes:

Tomar en cuenta que: LDPV - LEDs paso vehículo, LDAV - LEDs aviso vehículo, LDVD - LEDs vehículo detenido, LDPP - LEDs paso peaton y LDPD - LEDs peatón detenido.

Patrón de luces	LDVD	LDAV	LDPV	LDPD	LDPP
0	OFF	OF	ON	ON	OFF
1	OFF	OF	INTERMINTENTE	ON	OFF
2	OFF	ON	OFF	ON	OFF
3	ON	OF	OFF	OFF	ON
4	ON	OF	OFF	OFF	INTERMITENTE
5	ON	OF	OFF	ON	OFF

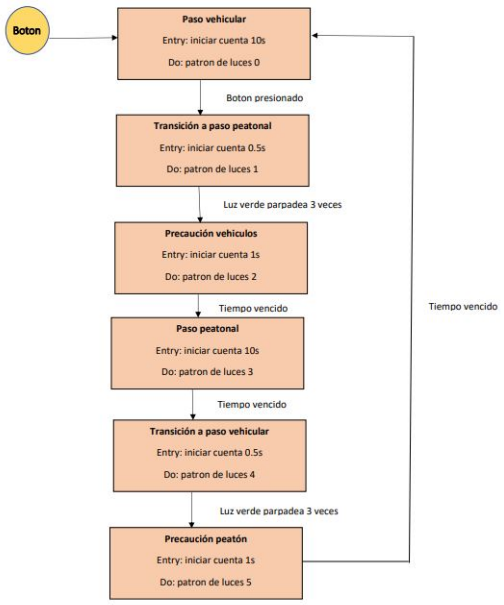

Figura 2: Diagrama de flujo de los estados del sistema.

3.2. Diagramas de tiempo resultantes

En la figura de la imagen 3 se puede ver el funcionamiento del semáforo, donde se puede ver que al hacer la solicitud, la luz verde del semáforo de los vehículos empieza a parpadea y después el semáforo de los vehículos pasa un momento a amarillo para indicar que desaceleren para por último pasar a rojo, donde los vehículos se detienen y el semáforo de los peatones se pone en verde. Después de un tiempo el semáforo peatonal empieza a parpadear para pasar de verde a rojo y el semáforo de los vehículos pasa de rojo a verde.

Luego en la figura de la imagen 4, podemos ver como se hace una solicitud con el botón del paso peatonal pero como el semáforo vehicular lleva menos de 10 segundos puesto en verde, entonces el semáforo no cambia inmediatamente, ya cuando pasa este tiempo mínimo entonces el semáforo sí empieza el proceso para dar el paso al peatón.

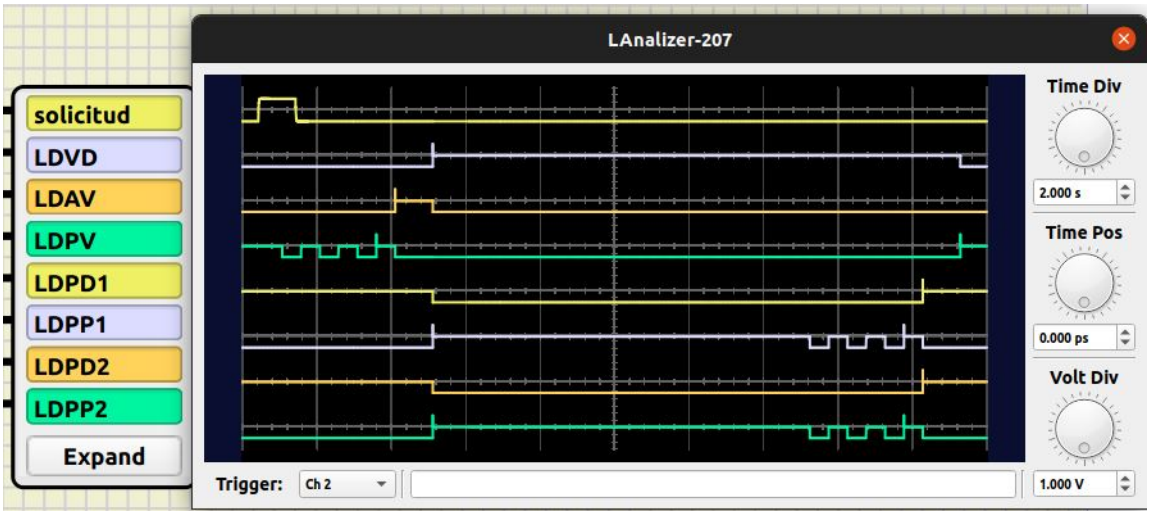

Figura 3: Diagrama de flujo de los estados del sistema.

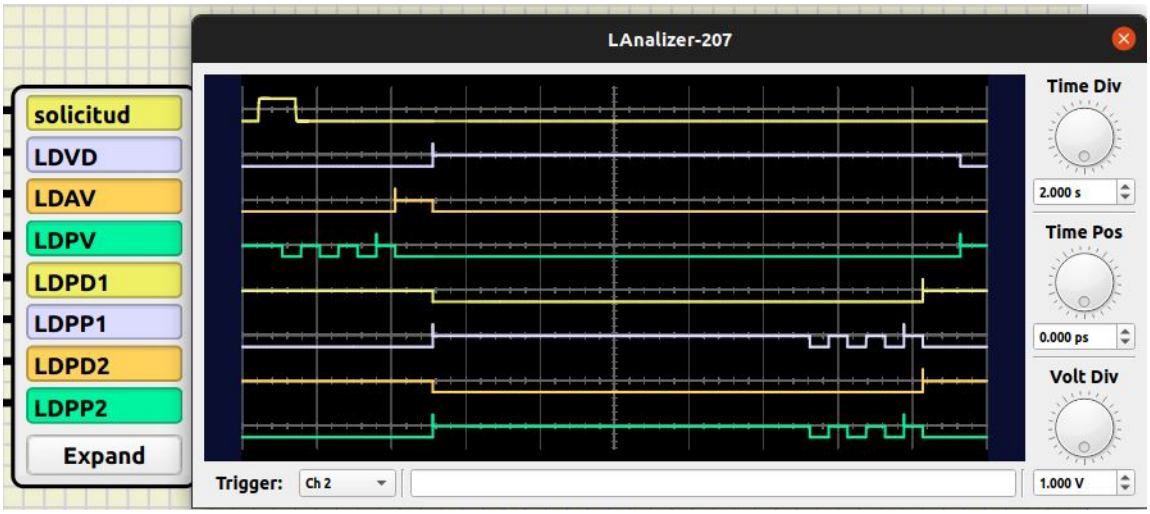

Figura 4: Diagrama de tiempos donde se muestra la espera del tiempo mínimo en color verde.

4. Enlace del repositorio

Enlace al repositorio: https://github.com/gokman26/Laboratorio-de-Microcontroladores.git

5. Conclusiones y recomendaciones

- Se logró desarrollar habilidades para controlar las entradas y salidas de un microcontrolador ATtiny4313 mediante el uso de código en C. Por otro lado, también se aprendió a trabajar con temporizadores internos del microcontrolador. Se implementó una rutina de interrupciones ante el cambio de una entrada del circuito, que permite hacer alguna variación a registros del microcontrolador.
- Se recomienda revisar a profundidad la hoja del fabricante del microcontrolador antes de trabajar con este, ya que este microcontrolador tiene una gran cantidad de registros que pueden dar mucha funcionalidad.
- Otra recomendación muy importante para tener en cuenta a la hora de trabajar con proyectos tan complejos es utilizar un programa de control de versiones como lo es GIT, ya que aveces se consiguen resultados deseados y por no guardar la versión funcional, posteriormente se le cambia algo al código y/o circuito y los resultados cambian.

6. Referencias

Referencias

[1] Atmel, "8-bit avr microcontroller with 2/4k bytes in-system programmable flash," 2011.