

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0624: Laboratorio de Microcontroladores

2 ciclo 2022

Laboratorio 2

GPIO, timers y FSM

Martin Sander Hangen
B97333

Jimmy Manley Peña
A73725

Profesor: Marco Villalta Fallas

18 de setiembre del 2022

Resumen

En el siguiente laboratorio se experimenta con el uso de interrupciones y temporizadores ayudándose del microcontrolador ATtiny4313 de Atmel. Se busca diseñar un semáforo y controlar un tipo de máquina de estados con el ATtiny4313. Así, las interrupciones de un temporizador o interrupciones externas con hardware serán de utilidad para interactuar con los estados del semáforo.

Al final del laboratorio se concluye que las interrupciones son de gran ayuda para realizar diferentes tareas que ocurren en momentos específicos durante el funcionamiento de este semáforo o máquina de estados. Esto porque una interrupción se puede dar en cualquier momento y es más eficiente programarlas. Además, se aprende que programar una máquina de estados no es una tarea increíblemente complicada y se puede lograr de manera efectiva con C.

Link del repositorio: https://github.com/martinsander00/lab_micros/tree/main/labs/lab2

1. Nota teórica

El microcontrolador que se utilizó durante la elaboración del laboratorio fue el ATtiny4313 con arquitectura AVR de Atmel. Este es de 8 bits y trabaja a una potencia baja. Hace uso de aproximadamente 18 pines que pueden estar en I/O y opera entre 1.8 V como mínimo y 6 V como máximo. El ATtiny4313 tiene dos temporizadores con cuatro canales PWM y también con prescalers. [3]

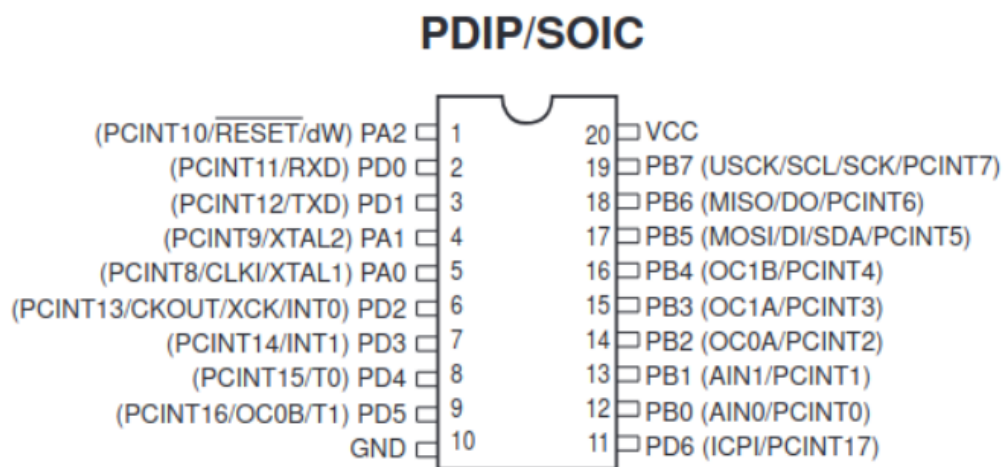


Figura 1. Diagrama del microcontrolador ATtiny4313. [3]

Se muestra también el diagrama de bloques a continuación:

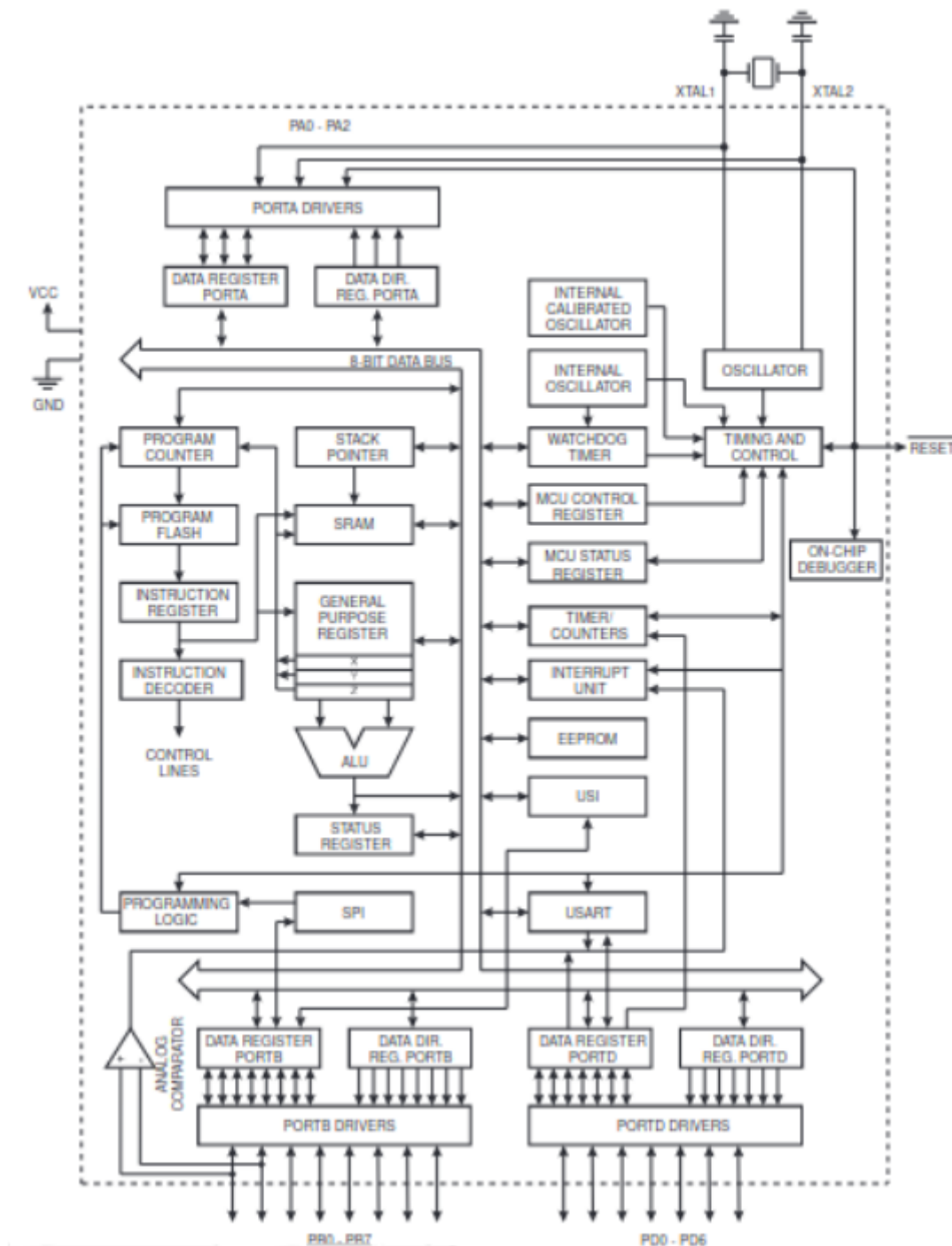


Figura 2. Diagrama de bloques del microcontrolador ATtiny4313. [3]

Durante el laboratorio se modificó la entrada MCUCR con el fin de que la interrupción 0 se activará en alto y la interrupción 1 en bajo. Con GIMSK se habilitaron ambas interrupciones externas.

Características generales

El ATtiny4313 es un microcontrolador CMOS de 8 bits de bajo consumo basado en la arquitectura RISC mejorada AVR. Al ejecutar instrucciones en un solo ciclo de reloj, el ATtiny4313 alcanza un rendimiento de hasta 1 MIPS por MHz permitiendo al diseñador de sistema optimizar el consumo de energía versus la velocidad de procesamiento. [3]

El AVR combina un set de instrucciones con 32 registros de trabajo de propósito general, Todos los 32 registros están directamente conectado con la Unidad Lógica Aritmética (ALU), permitiendo que dos registros independientes sean accedidos en una sola instrucción ejecutada en un solo ciclo de reloj. El resultado es una arquitectura que es más eficiente a nivel de código mientras alcanza un rendimiento hasta diez veces más rápido que microcontroladores CISC convencionales. [3]

Memoria

Dispositivo	Flash	EEPROM	RAM
ATtiny4313	4k bytes	256 Bytes	256 Bytes

Tabla 1. Memoria.

Para maximizar el desempeño y el paralelismo, AVR hace uso de una arquitectura de Harvard con buses y memorias separadas para datos e instrucciones. Las instrucciones en el programa son ejecutadas con un pipeline de un solo nivel. Mientras una instrucción es ejecutada, la siguiente instrucción es pre-fetched de la memoria de instrucciones. Esto permite que las instrucciones sean ejecutadas cada ciclo. La memoria del programa (las instrucciones) es memoria flash reprogramable en el sistema.

El registro de acceso rápido contiene 32 x 8-bit registros de propósito general con un tiempo de acceso de un ciclo de reloj. Esto permite operaciones de la ALU de un solo ciclo, dos operadores son sacados del archivo de registro, la operación es ejecutada, y el resultado es guardado de vuelta en el archivo de registro, todo en un ciclo. [3]

Durante las interrupciones o el llamado a subrutinas, el contador de programa de la dirección de retorno es guardado en el Stack. El Stack está efectivamente localizado en la SRAM de datos generales, y consecuentemente el Stack sólo está limitado por el tamaño total de la SRAM y el uso de la SRAM. Todos los programas de usuario deben de inicialar el SP (Stack Pointer) en la rutina de reset (antes que las subrutinas o las interrupciones sean ejecutadas). El SP es de libre lectura/escritura en el espacio de E/S. La SRAM de datos puede ser accedida fácilmente a través de cinco modos de direccionamiento distintos soportados en la arquitectura AVR.

El espacio de memoria en la arquitectura AVR es lineal. [3]

Un módulo flexible de interrupciones tiene su registro de control en el espacio de E/S con un bit de habilitación de Interrupciones Globales adicional en el registro de Estatus. Todas las interrupciones tienen un vector de interrupciones por separado en la tabla de Vector de Interrupciones. Las interrupciones tienen prioridad de acuerdo a su posición en el Vector de Interrupciones. Entre más baja sea la dirección del vector de interrupciones, mayor prioridad.

El espacio de E/S contiene 64 direcciones para las funciones de los periféricos del CPU como Registros de Control, y otras funciones de E/S. La memoria de E/S puede ser accedida directamente, o como locaciones en el Espacio de datos, siguiendo las del Archivo de registros, 0x20 – 0x5F.

1.2. Componentes electrónicos adicionales

Dos botones “push” o pulsadores fueron los desencadenantes del problema por resolver. Cada uno con resistores de 10k Ohm y un capacitor en paralelo para evitar oscilaciones.

Se hizo uso de luces LED rojas y verdes en forma de diodos. En total se necesitaron 6 LEDs. Los diodos rojos utilizaban un voltaje de 2.5 V y los verdes de 3.2 V máximo [3]. Con cada uno de estos diodos se tenía una resistencia de 120 Ohm en serie.

1.3. Diseño del circuito

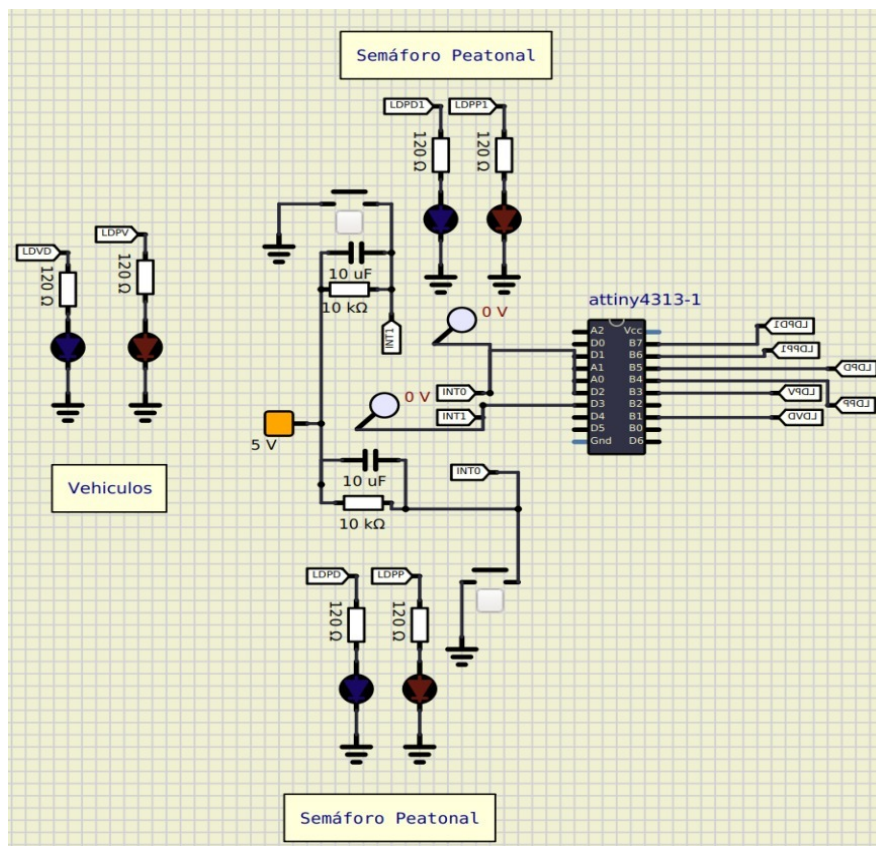


Figura 3. Diseño del circuito

Para los resistores en serie con los LEDs se eligió una magnitud de resistencia con la cual se pudiera tener una corriente en cada uno de los LEDs de 15mA a 20mA. Como la hoja del fabricante explica, esto brindaría una iluminación suficiente. Se toma un voltaje directo máximo en los LEDs rojos de 2.5 V [4]. Con LTK se tiene que:

$$5V = V_R + 2,5 V \longrightarrow V_R = 2,5 V$$

Ecuación 1. LTK para LED rojo.

Ahora, se tiene que cuando el diodo recibe tensión directa del resistor el voltaje es de 2.5 V. Entonces para que no se exceda del valor deseado se calcula la resistencia con la Ley de Ohm:

$$20 \text{ mA} = \frac{2,5}{R} \longrightarrow R = 125\Omega$$

Ecuación 2. Ley de Ohm para LED rojo.

Se toma el valor 120 Ohm como valor aproximado entonces.

Ahora para los dos botones, se tiene una configuración pull-up que consiste en estar en alto cuando no se presiona el botón. Por el contrario, cuando se presiona el botón se baja el nivel. Se escoge una resistencia de 100k Ohm con el fin de lograr que la caída de voltaje esté en la resistencia de pull-up. Para ello se necesita que la resistencia de pull-up sea mucho mayor que la del botón.

Cuando se presione el botón, la tensión en la terminal del pin del controlador que está conectada a tierra baja será casi 0. Por otro lado, cuando no se esté presionando el botón se tiene un circuito abierto entonces la resistencia será de 0 V exactamente y la tensión en el pin va a ser de 5 V porque no se tienen más caídas.

Como se usó en el laboratorio anterior, el portal Sparkfun es de gran ayuda para averiguar el precio de los componentes utilizados durante el laboratorio. Los LEDs rojos y verdes de 5 mm tienen un precio de 0.45 USD. Las resistencias cuestan 0.09 USD por unidad y se usaron en total 8. Los botones van a costar 0.14 USD por unidad y el ATtiny 4313 en Atmel está a un precio de 1.68 USD.

Componente	Cantidad	Precio
Resistores	8	\$0.72
Botones	2	\$0.28
LEDs	6	\$2.7
ATtiny4313	1	\$1.68
Total	-	\$5.38

Tabla 2. Precio del circuito completo.

2. Desarrollo/Análisis de resultados

El desarrollo de la función del semáforo se implementa mediante una máquina de estados finitos (FSM). En la cual se tienen los siguientes estados:

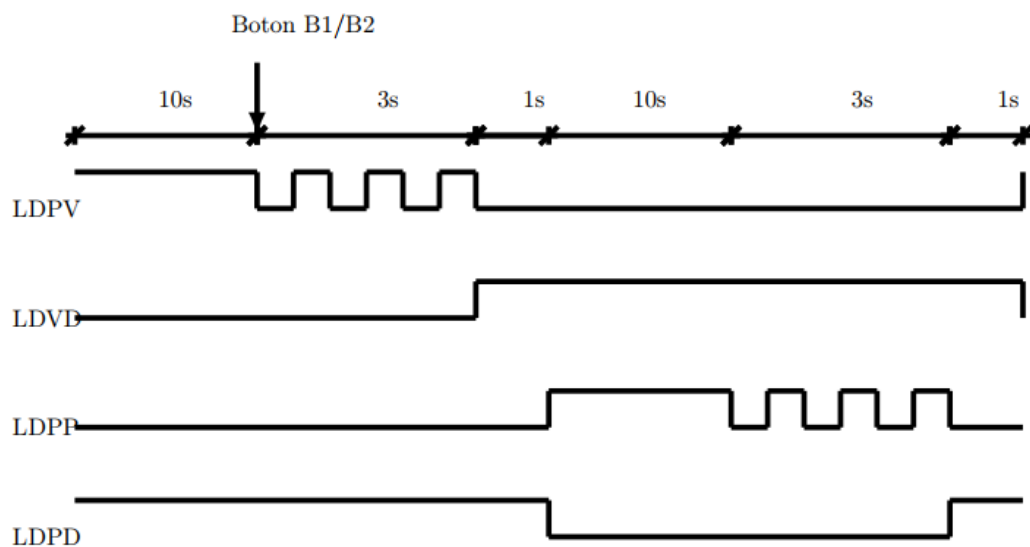


Figura 2: Diagrama de temporización

- **LDPP:** paso de peatones.
- **LDVD:** vehículos detenidos.
- **LDPD:** peatones detenidos.

Las funciones específicas del ATtiny4313 utilizadas son:

- Las interrupciones INT0 e INT1 para los botones que activan el paso peatonal.
- EL temporizador de 8 bits.

Desarrollo de las interrupciones y el temporizador para lograr la FSM requerida.

Cada uno de los estados debe de esperar cierto tiempo y en los casos donde se da una transición entre el paso peatonal y el paso vehicular los leds del estado que va cambiar deben de parpadear durante 3 segundos.

Toda esta información se resume en la Figura 5.

La función más importante es la del temporizador que se encarga de contar el tiempo necesario que cada estado debe de durar. Para lo cual se obtiene el número de veces que el temporizador de 8 bits cuenta de 0 a 255 en un segundo. Utilizando el preescaler de 1024, se tiene una frecuencia f de 15625, de lo cual se obtiene que el contador cuenta un total de 61 veces de 0 a 255 durante 1 segundo.

Con este valor de la cantidad de overflows para el contador, ya tenemos como medir un segundo. Por lo que a la función de interrupción por overflow se llama cada vez que ocurre un overflow y se comprueba si ha pasado la cantidad de tiempo requerida. Estando la cantidad de tiempo requerida dada por la variable delay, como sabemos 60 representa 1 s, por lo que 30 representaría medio segundo y de esta forma podemos obtener cualquier tiempo requerido.

Luego de esto se crea una función fsm, que es la máquina de estados la cual cumple las transiciones establecidas en la Figura 5.

2.1. Funcionalidad del programa

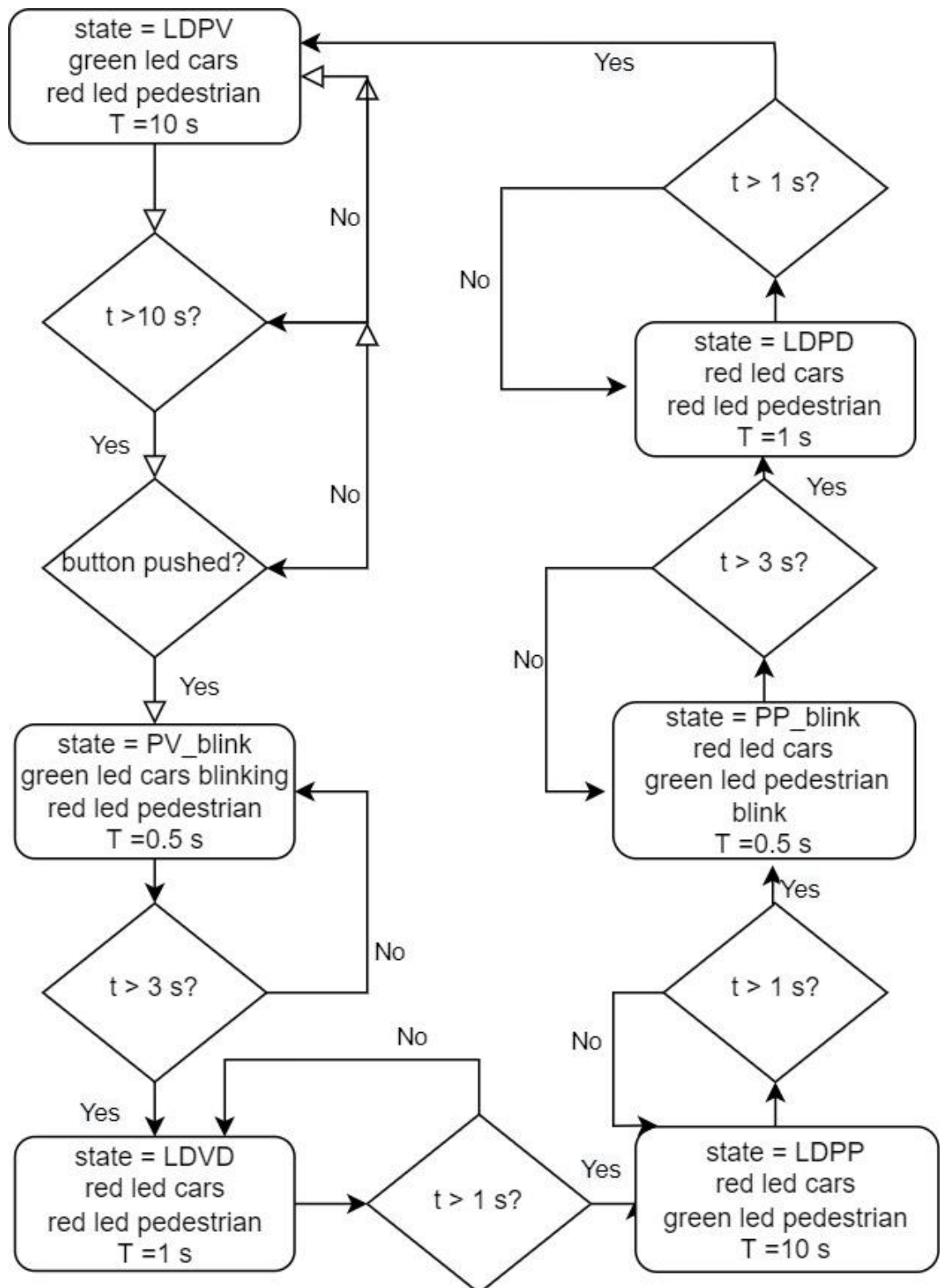


Figura 5. Diagrama de estados

2.2. Funcionalidad electrónica

En la siguiente figura se logra apreciar el funcionamiento de las resistencias pull-up. Donde cuando los botones no se presionan INT0 e INT1 son iguales a 5 V. En la figura también se aprecia cómo se enciende en rojo la luz de los semáforos peatonales en rojo lo cual comprueba el funcionamiento de los LEDs rojos y de sus resistencias de 120 Ohm.

Para comprobar cómo se logró el parpadeo de los LEDs explicado en la sección anterior en código, se usó el Oscopio-86 el cual es un osciloscopio normal. Este muestra una señal amarilla y verde. La señal amarilla corresponde a la luz del semáforo vehicular en verde que parpadea por aproximadamente 3 s para seguidamente ponerse en rojo.

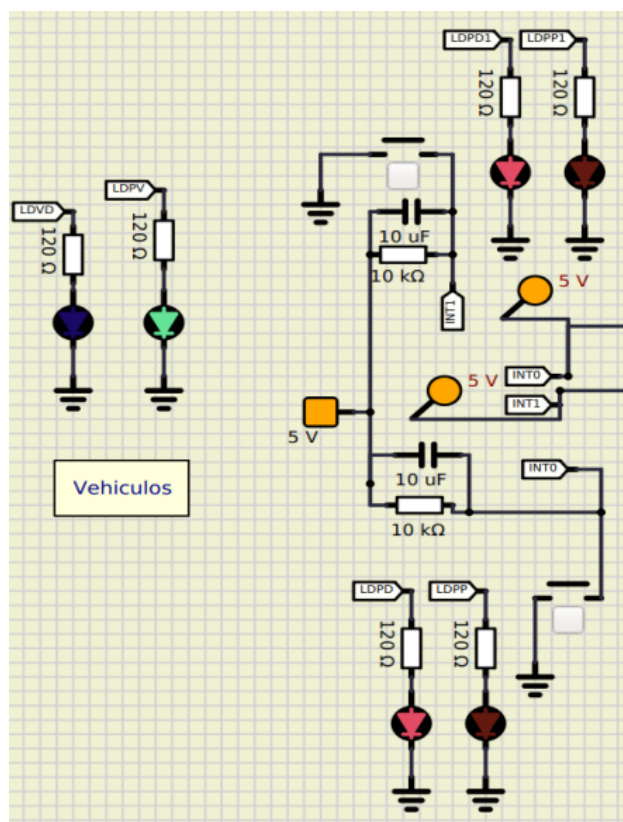


Figura 6. Resistores pull-up.

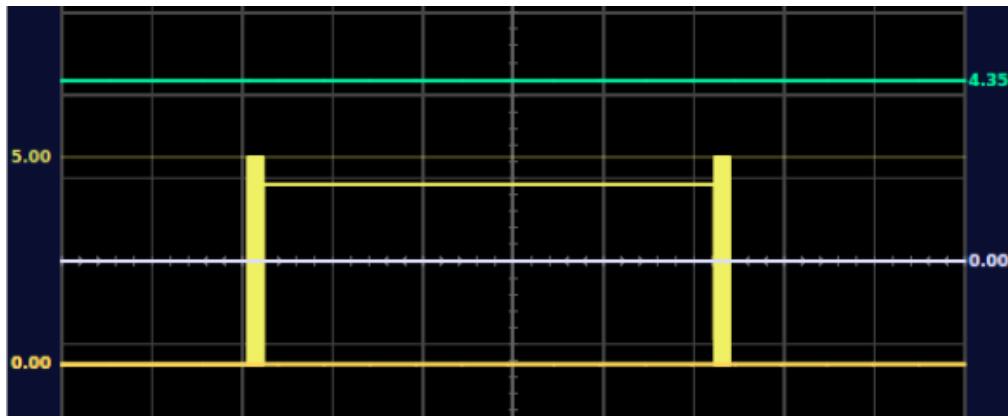


Figura 7: Parpadeo de 3 s del semáforo vehicular tras presionar botón.

Finalmente, se tiene en la figura 7 dos imágenes. La de la izquierda enseña el funcionamiento del circuito después de que ya se presionó el botón. Los LEDs verdes de los semáforos peatonales se encienden y el LED rojo del semáforo vehicular se enciende. Aparte, se tiene en la imagen de la derecha el funcionamiento contrario. LED verde del semáforo vehicular encendido y LEDs rojas de los semáforos peatonales encendidos.

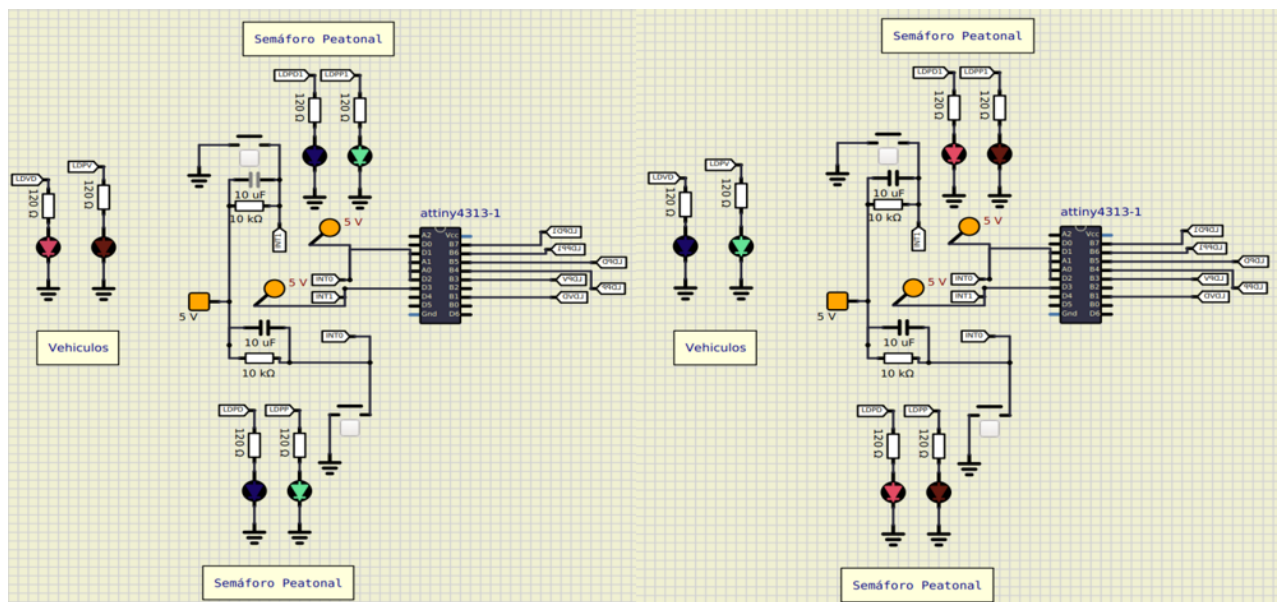


Figura 8. Circuito prendido y circuito apagado.

3. Conclusiones y recomendaciones

Las interrupciones permiten medir eventos fuera de la ejecución normal del programa. Si se utilizaran delays, que no son eventos de interrupción, no se tendría la confianza de que un evento, como presionar el botón del paso peatonal, vaya a ser atendido de forma inmediata una y otra vez. Para aplicaciones que requieren de eventos que deben ser atendidos inmediatamente las interrupciones son la mejor opción.

La máquina de estados nos permitió codificar el funcionamiento del semáforo de una forma fácil pero como recomendación se podrían explorar otras opciones como lo son los structs/enums, las lookup tables o con punteros a funciones.

Referencias

- [1] Microchip, PIC12F675. Microchip, 1 ed., 2003.
- [2] SimulIDE. Recuperado de: <https://www.simulide.com/p/seven-segment-display.html>
- [3] Atmel, ATtiny2313A/ATtiny4313. 8246B, 1 ed., 2011.
- [4] Multicomp, Standard Red-emitting LED. Farnell, 1.1 ed., 2011.
- [5] Sparkfun, LED Basic Red 5mm. Sparkfun Electronics, s.f.
<https://www.sparkfun.com/products/9590>.
- [5]https://ww1.microchip.com/downloads/en/Appnotes/Atmel-2505-Setup-and-Use-of-AVR-Timers_ApplicationNote_AVR130.pdf

Apéndice

Green LED - T1 (5mm)



RoHS
Compliant

Features

- Standard 5mm round package
- High luminous output
- Water clear lens

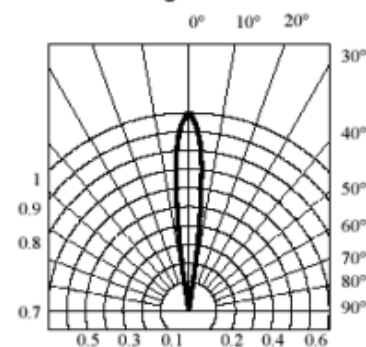
Maximum Ratings at T_A = 25°C

Reverse Voltage (<100µA)	: 5V
D.C. Forward Current	: 30mA
Pulse Current (Pulse Width of 0.1ms, 1/10 Duty Cycle)	: 100mA
Operating Temperature Range	: -25°C to +85°C
Storage Temperature Range	: -40°C to 100°C
Soldering Temperature Dip Soldering	: 260°C for 5s
Soldering Temperature Hand Soldering	: 350°C for 3s

Electrical and Optical Characteristics at T_A = 25°C

LED Chip			Lens Colours	Dominant Wavelength (nm) at 20mA	Luminous Intensity (mcd) at 20mA		Forward Voltage (V) at 20mA		Viewing Angle 2θ ^{1/2} (°)
Material	Emitted Colours	Brightness			Minimum	Typical	Typical	Maximum	
InGaN / Sapphire	True Green	Mega	Water Clear	520	19,500	45,000	3.2	4	15

Radiation Diagrams



www.element14.com
www.farnell.com
www.newark.com



Standard LED

Red Emitting Colour



Absolute Maximum Ratings at $T_a = 25^{\circ}\text{C}$

Parameter	Maximum	Unit
Power Dissipation	80	mW
Peak Forward Current (1/10 Duty Cycle, 0.1 ms Pulse Width)	100	mA
Continuous Forward Current	20	
Derating Linear From 50°C	0.4	mA / $^{\circ}\text{C}$
Reverse Voltage	5	V
Operating Temperature Range	-25°C to $+80^{\circ}\text{C}$	
Storage Temperature Range	-40°C to $+100^{\circ}\text{C}$	
Lead Soldering Temperature (4 mm (0.157) Inches from Body)	260 $^{\circ}\text{C}$ for 5 s	

Electrical Optical Characteristics at $T_a = 25^{\circ}\text{C}$

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Test Condition
Luminous Intensity	I_v		40		mcd	$I_f = 20 \text{ mA}$ (Note 1)
Viewing Angle	$2\theta_{1/2}$		25		Deg	(Note 2)
Peak Emission Wavelength	λ_p		640		nm	$I_f = 20 \text{ mA}$
Dominant Wavelength	λ_d		635		nm	$I_f = 20 \text{ mA}$ (Note 3)
Spectral Line Half-Width	$\Delta\lambda$		25		nm	$I_f = 20 \text{ mA}$
Forward Voltage	V_f		2	2.5	V	$I_f = 20 \text{ mA}$
Reverse Current	I_R	-	-	100	μA	$V_R = 5 \text{ V}$

Notes:

1. Luminous intensity is measured with a light sensor and filter combination that approximates the CIE eye-response curve
2. $\theta_{1/2}$ is the off-axis angle at which the luminous intensity is half the axial luminous intensity
3. The dominant wavelength (λ_d) is derived from the CIE chromaticity diagram and represents the single wavelength which defines the colour of the device

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 120 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
- Data and Non-volatile Program and Data Memories
 - 2/4K Bytes of In-System Self Programmable Flash
 - Endurance 10,000 Write/Erase Cycles
 - 128/256 Bytes In-System Programmable EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 128/256 Bytes Internal SRAM
 - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
 - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes
 - Four PWM Channels
 - On-chip Analog Comparator
 - Programmable Watchdog Timer with On-chip Oscillator
 - USI – Universal Serial Interface
 - Full Duplex USART
- Special Microcontroller Features
 - debugWIRE On-chip Debugging
 - In-System Programmable via SPI Port
 - External and Internal Interrupt Sources
 - Low-power Idle, Power-down, and Standby Modes
 - Enhanced Power-on Reset Circuit
 - Programmable Brown-out Detection Circuit
 - Internal Calibrated Oscillator
- I/O and Packages
 - 18 Programmable I/O Lines
 - 20-pin PDIP, 20-pin SOIC, 20-pad MLF/VQFN
- Operating Voltage
 - 1.8 – 5.5V
- Speed Grades
 - 0 – 4 MHz @ 1.8 – 5.5V
 - 0 – 10 MHz @ 2.7 – 5.5V
 - 0 – 20 MHz @ 4.5 – 5.5V
- Industrial Temperature Range: -40°C to +85°C
- Low Power Consumption
 - Active Mode
 - 190 µA at 1.8V and 1MHz
 - Idle Mode
 - 24 µA at 1.8V and 1MHz
 - Power-down Mode
 - 0.1 µA at 1.8V and +25°C



8-bit AVR[®]
Microcontroller
with 2/4K Bytes
In-System
Programmable
Flash

ATtiny2313A
ATtiny4313

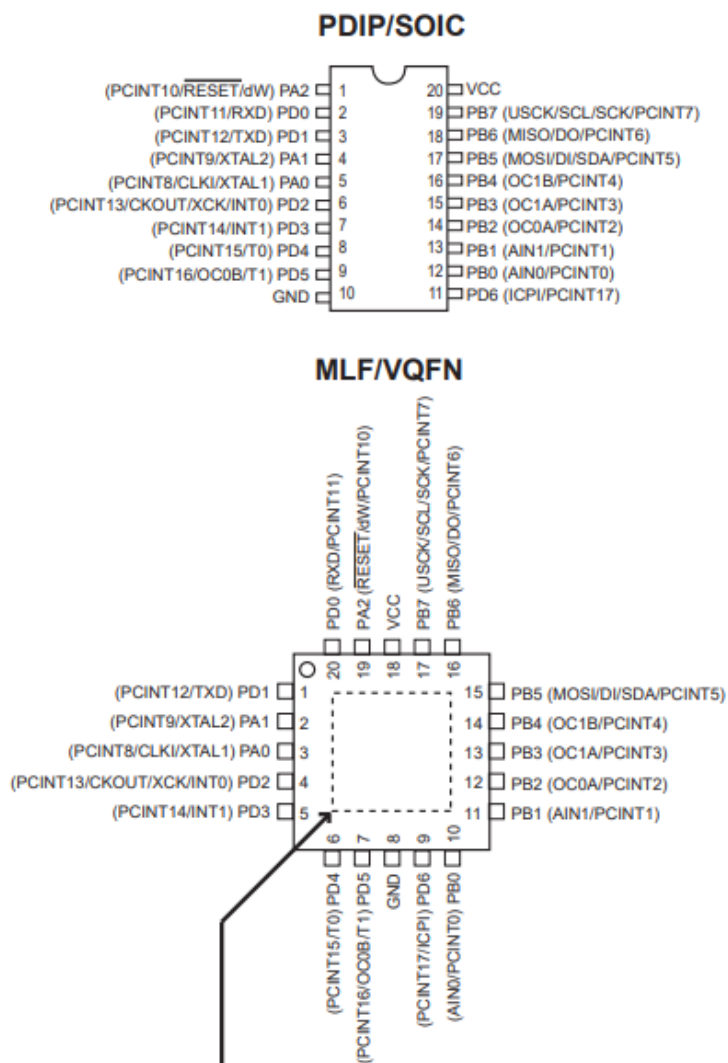
Rev. 8246B-AVR-09/11





1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313



2. Overview

The ATtiny2313A/4313 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny2313A/4313 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram

