Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624: Laboratorio de Microcontroladores
I ciclo 2022

Laboratorio 2

GPIO, interrupciones, timers y el ATtiny4313

Matías Leandro Flores
B94199

Profesor: Marco Villalta Fallas

12 de mayo de 2022

Resumen

El presenta laboratorio aborda el uso de temporizadores e interrupciones por hardware del microcontrolador ATtiny4313 de Atmel, mediante el diseño de un semáforo. El semáforo diseñado con el ATtini4313 se realizó mediante una máquina de estados finita, la cual, con el uso de interrupciones del temporizador e interrupciones externas y sus rutinas de atención, permite cambiar los estados del semáforo.

Con la elaboración del Laboratorio 2: GPIO, interrupciones, timers y el ATtiny4313 se concluyó que realizar ciertas funciones que pueden ocurrir en cualquier momento del funcionamiento resulta más eficiente con interrupciones, ya que se pueden ejecutar en cualquier momento del programa y resultan más rápidas que programar algo dentro de la función principal. También se concluyó que el concepto de máquina de estados finita se puede propagar a un paradigma de programación, lo que permite tener una mejor estructura de la solución a implementar.

- El repositorio del presente laboratorio se puede acceder en el siguiente enlace: https://github.com/matflow/Labo2_Microcontroladores.git.

1. Nota teórica

1.1. Microcontrolador ATtiny4313

El microncontrolador utilizado para la implementación del dado es el ATtiny4313 de arquitectura AVR de Atmel. Este microcontrolador de 8 bits es de baja potencia y cuenta con 18 pines de I/O y un rango de voltajes de operación de 1.8 a 5 V. Además, entre sus periféricos cuenta con dos temporizadores con prescalers separados y cuatro canales PWM [1]. Su diagrama de pines se presenta en la Fig. 1, mientras que su diagrama de bloques está en la Fig. 2.

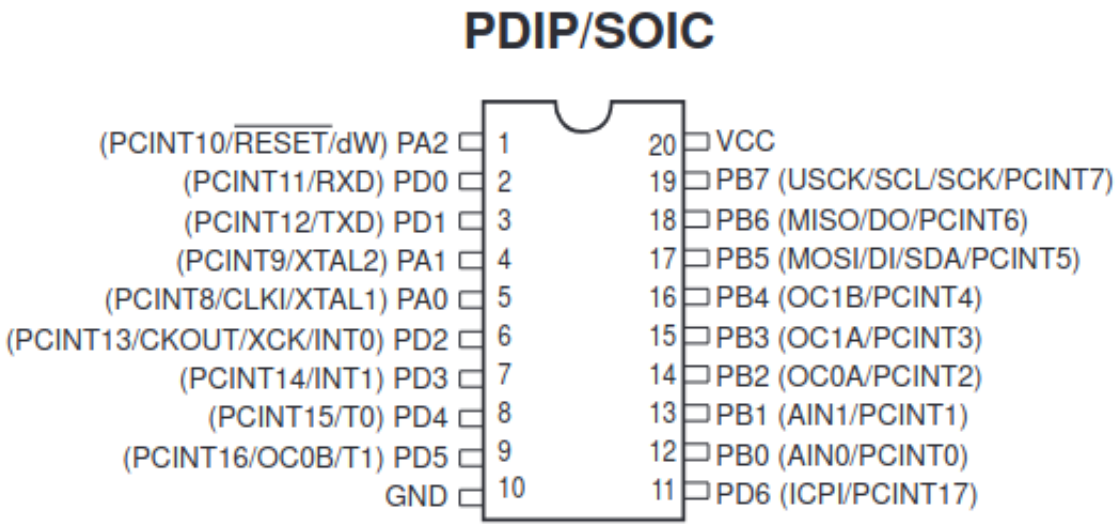


Figura 1: Diagrama de pines del ATtiny4313 [1]

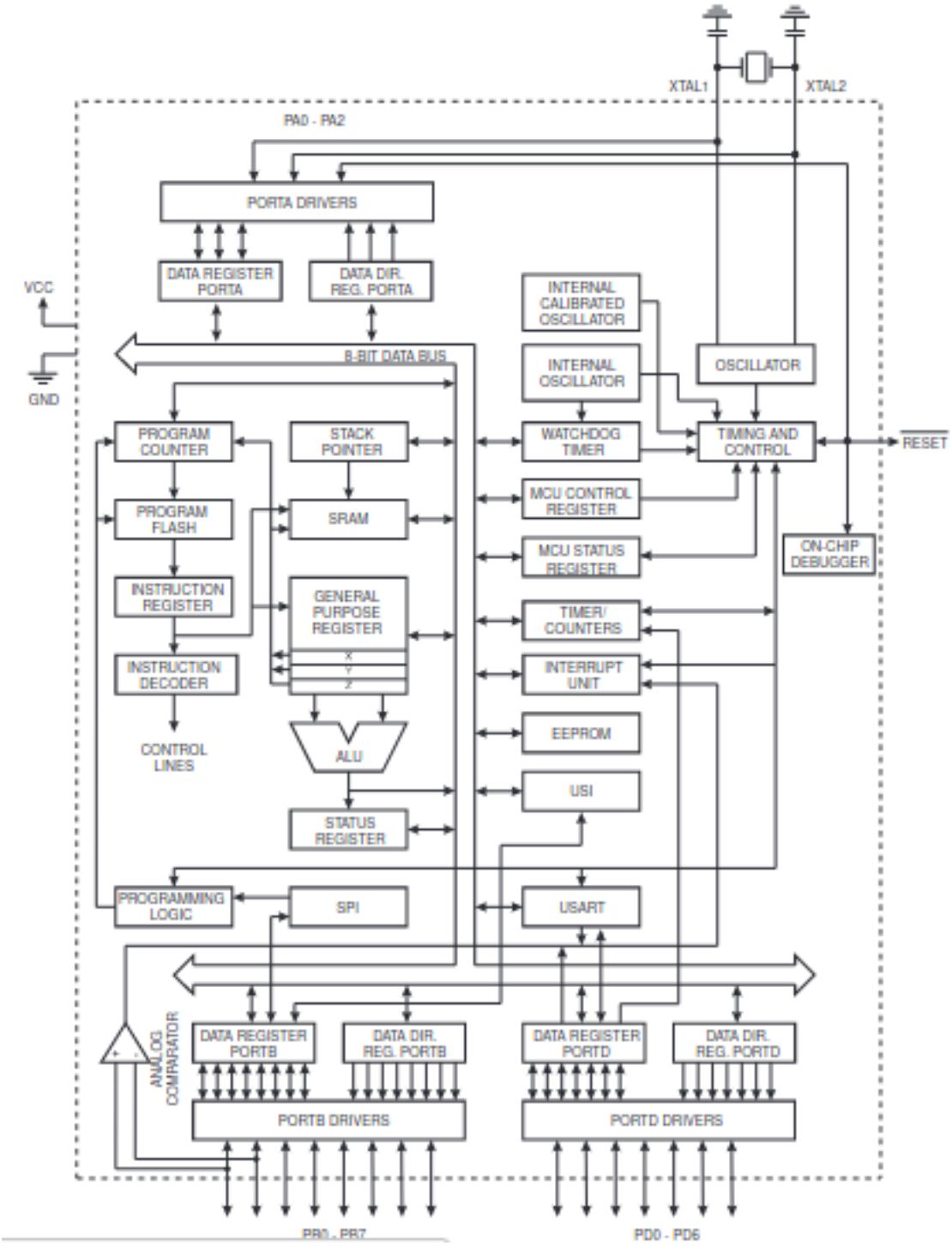


Figura 2: Diagrama de pines del ATtiny4313 [1]

Entre sus rangos máximos absolutos y características eléctricas está una temperatura de operación de -55 °C a +125 °C, un voltaje máximo de operación de 6 V y una corriente DC máxima por pin I/O de 40 mA [1]. Además, su voltaje de umbral bajo va entre $V_{IL} = -0{,}5 - 0{,}2V_{cc}$ y su voltaje de umbral alto puede estar entre $V_{IH} = 0{,}7V_{cc} - Vcc + 0{,}5$.

Los periféricos utilizados con este microcontrolador fueron el temporizador de 8 bits *timer0* que

se configuró con los registros TCCR0A y TCCR0B para utilizar el modo normal de operación en el que se cuenta de 0 a 255 y luego se activa el tag de overflow, además de configurar el prescaler para utilizar una frecuencia del temporizador de $f/1024$. También se utilizaron las interrupciones externas de INT0 e INT1, habilitadas con el registro GIMSK y se modificó el registro MCUCR para que la interrupción INT0 se activara con el flanco creciente de la señal y INT1 con el flanco decreciente.

1.2. Componentes electrónicos adicionales

Como componentes electrónicos adicionales al microcontrolador se utilizaron **diodos LED rojos** y **diodos LED verdes**, para un total de seis LEDs. Se asumirá para los LEDs rojos que se utilizó el LED rojo estándar de *Multicomp* [2], con un voltaje en directo máximo de 2.5 V. y el LED verde T1 de *Multicomp* [3] con un voltaje en directo máximo de 3.2 V. También se utilizaron dos botones pulsadores simples, cada uno con **resistores de 10k** Ω pullup en paralelo con un **capacitor de 10 uF** para almacenar el nivel de voltaje y evitar oscilaciones mecánicas, además de seis resistores de 120 Ω en serie con cada diodo LED.

1.3. Diseño del circuito

Para el diseño de los resistores de 120 Ω en serie con cada LED se partió de desear una corriente en cada LED de entre 15mA a 20mA, lo cual aseguraría una buena iluminación. Para asegurar que no se sobrepase este valor de corriente se asumirá el nivel de voltaje en directo máximo del diodo rojo de Multicomp [2] de 2.5 V. De esta forma, realizando la ley de tensiones de Kirchoff en el lazo de un LED y el resistor a diseñar se tiene lo siguiente.

$$5V = V_R + 2{,}5\ V \longrightarrow V_R = 2{,}5\ V \tag{1}$$

O sea que, cuando el diodo está polarizado en directa la tensión del resistor a diseñar será de 2.5 V y para asegurar que no sobrepase los 20 mA:

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

Este valor se aproxima a $R = 120\Omega$ para los seis resistores de los LEDs de semáforos.

Para el diseño de la señal activada por botón se realizó una configuración de pull-up, lo que establece un nivel alto cuando el botón no está siendo presionado, y un nivel bajo cuando el botón es presionado. Para elegir la resistencia de pull-up, se debe asegurar que esta sea de un valor elevado para que al cerrar el circuito (presionar el botón) la gran mayoría de la caída de voltaje esté en el resistor de pull-up. Para asegurar que prácticamente todo el voltaje de alimentación esté en la resistencia de pull-up, la resistencia de pull-up debe ser mucho mayor que la resistencia interna del botón, por lo que se eligió un valor de **10 k**Ω. La terminal a tierra baja del resistor de pull-up está conectada a uno de los pines del microcontrolador, así cuando el botón está presionado el valor del voltaje del pin será cercano a 0 V y cuando el botón no está presionado habrá un circuito abierto por lo que la resistencia será 0 V y el voltaje del pin será de 5 V ya que no hay caídas por resistencia.

El esquemático en SimulIDE del circuito diseñado se presenta en la Fig. 3.

En cuanto a los precios, de acuerdo con el portal Sparkfun, se tiene un precio de $0.45 para un LED básico rojo de 5mm, se utilizará este precio para los LED verdes también [4]. En cuanto a los resistores, un precio común para resistores de 1 a 4 W individuales es de $0.09. Se encontró un precio

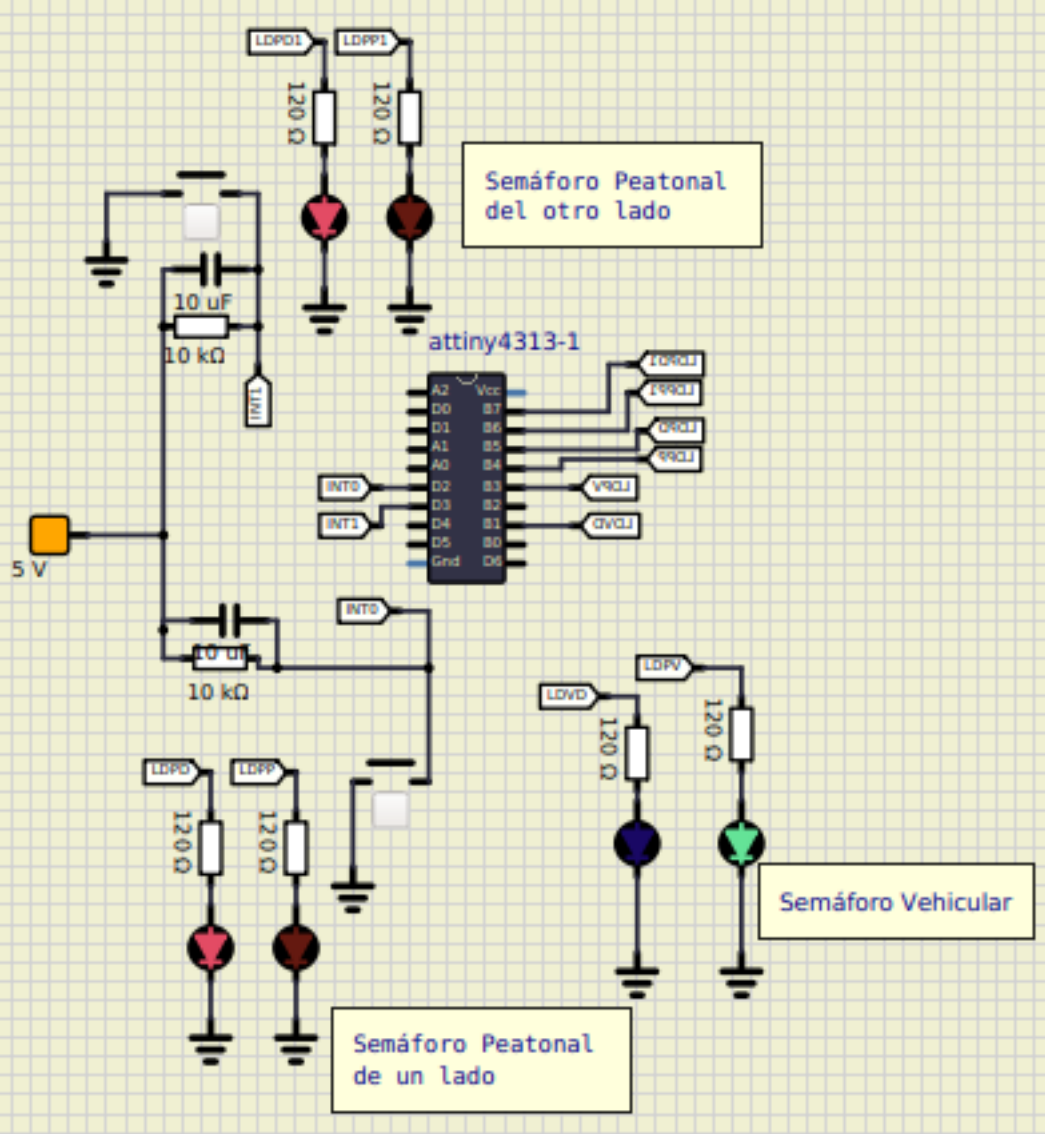


Figura 3: Esquemático en SimulIDE de circuito de semáforo diseñado.

de $6.88 para 50 botones pulsadores [5], lo que significa un precio de $0.14 por botón. Además, el precio actual del ATtiny4313 en el portal de Atmel es de $1.68. De esta forma se tiene el siguiente precio para el sistema de semáforo creado.

Componente	**Cantidad**	**Precio**
Resistores 10k y 120 Ω	8	$0.72
Botón	1	$0.14
Diodo LED	6	$2.7
ATtiny4313	1	$1.68
Total	-	$5.24

Tabla 1: Lista de componentes

2. Desarrollo/Análisis de resultados

2.1. Funcionalidad del programa

Para realizar la máquina de estados del semáforo utilizando interrupciones, fue importante establecer la cantidad de veces que el temporizador *timer0* cuenta de 0 a 255 (cantidad de overflows)para generar un segundo. Con una frecuencia del microcontrolador de 16 MHz el prescaler de $f/1024$ se tiene $f = 15625$, el procedimiento para determinar una espera de un segundo con el temporizador fue la siguiente.

$$T = 1/f = 1/15625 = 64\ us \tag{3}$$

$$T_{overflow} = 64\ us \cdot 255 = 16,32\ ms \tag{4}$$

$$Overflows = \frac{1\ s}{16,32\ ms} = 61{,}27 \tag{5}$$

Por lo tanto, deben ocurrir 61.27 cuentas de 0 a 255, o 61.27 overflows, lo que se redondea a 61. Sabiendo esto, en la rutina de interrupción de overflow del *timer0* se especificó lo siguiente:

```
ISR(TIMER0_OVF_vect){
    if(count == curr_delay){
        valid = 1;  // pasó al menos 1 t
        blinks++;   // cantidad de cambios de luz o #t
        count = 0;
    } else{
        count++;
        valid = 0;  // no cambiar luz
    }
}
```

Si se desea una espera de 1 segundo (t = 1s), entonces el valor de **curr_delay** es de 61, y si la cuenta llega a este valor se activa *valid*, que indica que ya pasó al menos 1t. Cada vez que se llega al valor de *curr_delay* se aumenta *blinks*, un entero que indica la cantidad de t's que ha pasado. Mediante la señal de valid se puede saber si ya se pasó el tiempo 't', ya sea este 1 segundo, 10 segundos, o el tiempo que sea que se especificó a *curr_delay*. Así entonces, la máquina de estados finita (FSM) diseñada utiliza la variable de *valid* para responder la pregunta: "¿Ya pasó el tiempo necesario?". El diagrama de la FSM se presenta en la Fig. 4.

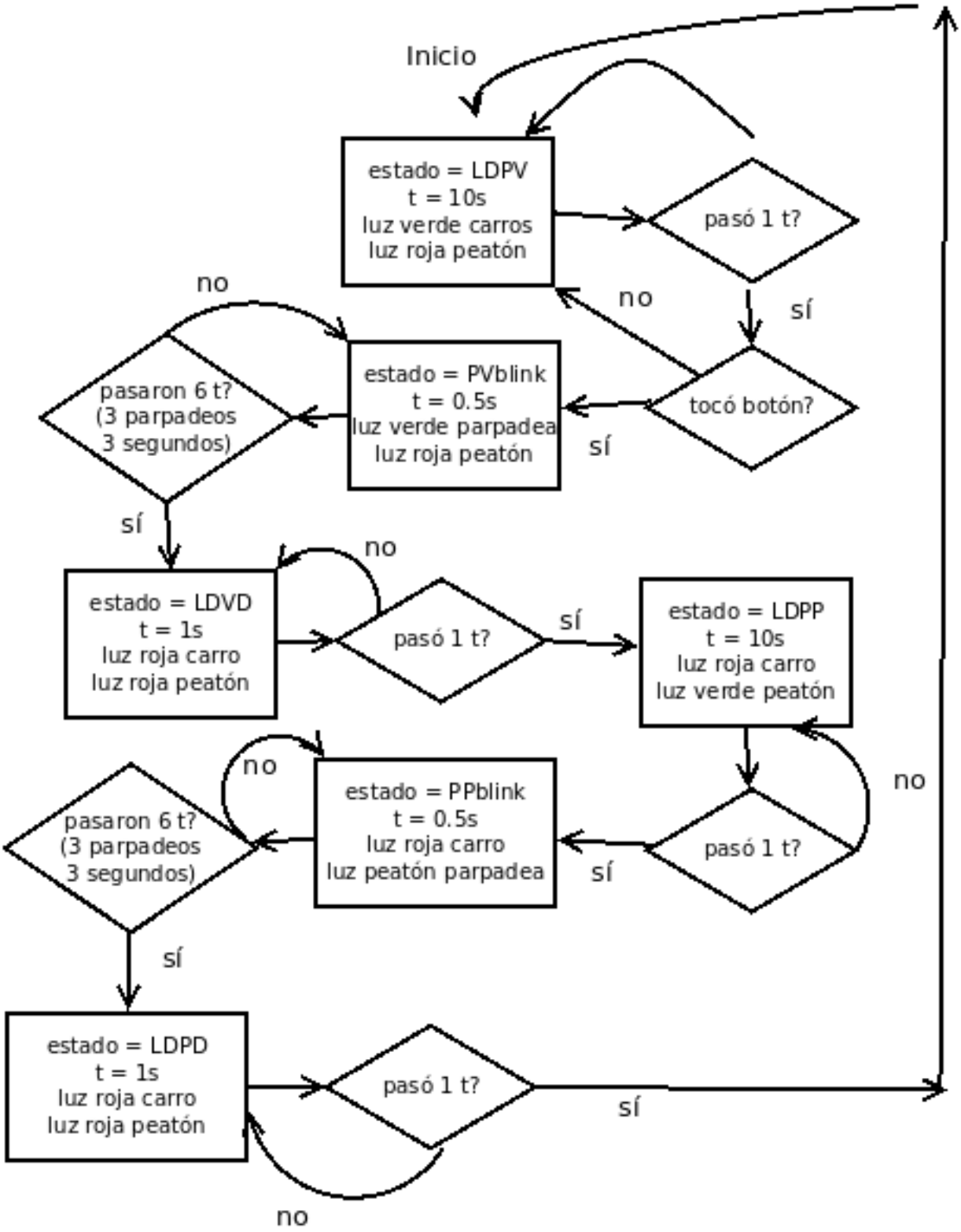


Figura 4: Diagrama de la máquina de estados finita implementada

La señal de *blink* de la rutina de atención de servicios permite parpadear los LEDs. Si se desea que una luz parpadee por 3 segundos, se especifica un t = 0.5s (curr_delay = 30) y por cada vez que

valid se pone en alto se cambia el LED, hasta que blink sea 6, lo que significa que hubieron 6 cambios (un parpadeo completo corresponde a dos cambios del LED).

Para la rutina de atención de servicio por interrupción externa, simplemente se especificó que ponga la señal de botón en alto solo si se está en el estado LDPV de semáforo de carros en verde, por lo que en cualquier otro momento presionar el botón no tiene un efecto. Luego de pasar del estado LDPV la señal de botón se vuelve a poner en 0, y solo se vuelve a activar hasta que se vuelva al estado LDPV y haya una interrupción externa por alguno de los dos botones. La misma rutina se utilizó para INT1_vect.

```
ISR(INT0_vect){
    if (estado == LDPV){
            boton = 1;
    }
}
```

2.2. Funcionalidad electrónica

La configuración de los resistores pull-up y el diseño de su valor de 10k Ω se verifica con la Fig. 5, donde se evidencia que con los botones no presionados el valor de los pines INT0 e INT1 es de 5 V.

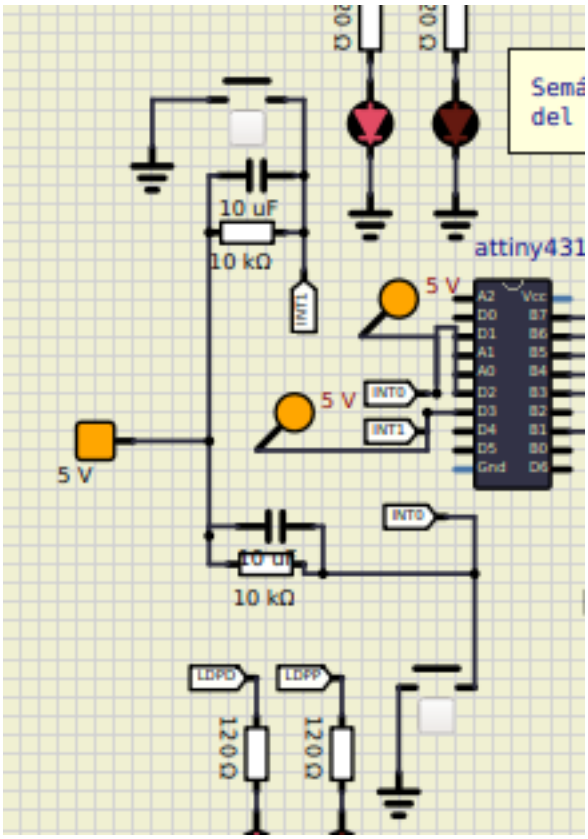


Figura 5: Funcionalidad de resistores pull-up

Además, en la parte superior de la Fig. 5 se evidencia la luz roja de un semáforo de peatón encendida, comprobando el valor de las resistencias de los LEDs de 120 Ω.

La funcionalidad del parpadeo de los LEDs se puede apreciar con la captura de la Fig. 6. La señal amarilla en dicha figura corresponde a la luz LDPV (luz verde de semáforo vehicular), que está parpadeando por tres segundos para luego cambiar a LDVD (luz roja vehicular). Del osciloscopio se muestran un flanco creciente de esta señal y luego un flanco decreciente, que se manifiesta como un parpadeo de dicho LED. En la esquina inferior de la figura se muestra el led LDPV, el cual está apagado en este instante del parpadeo.

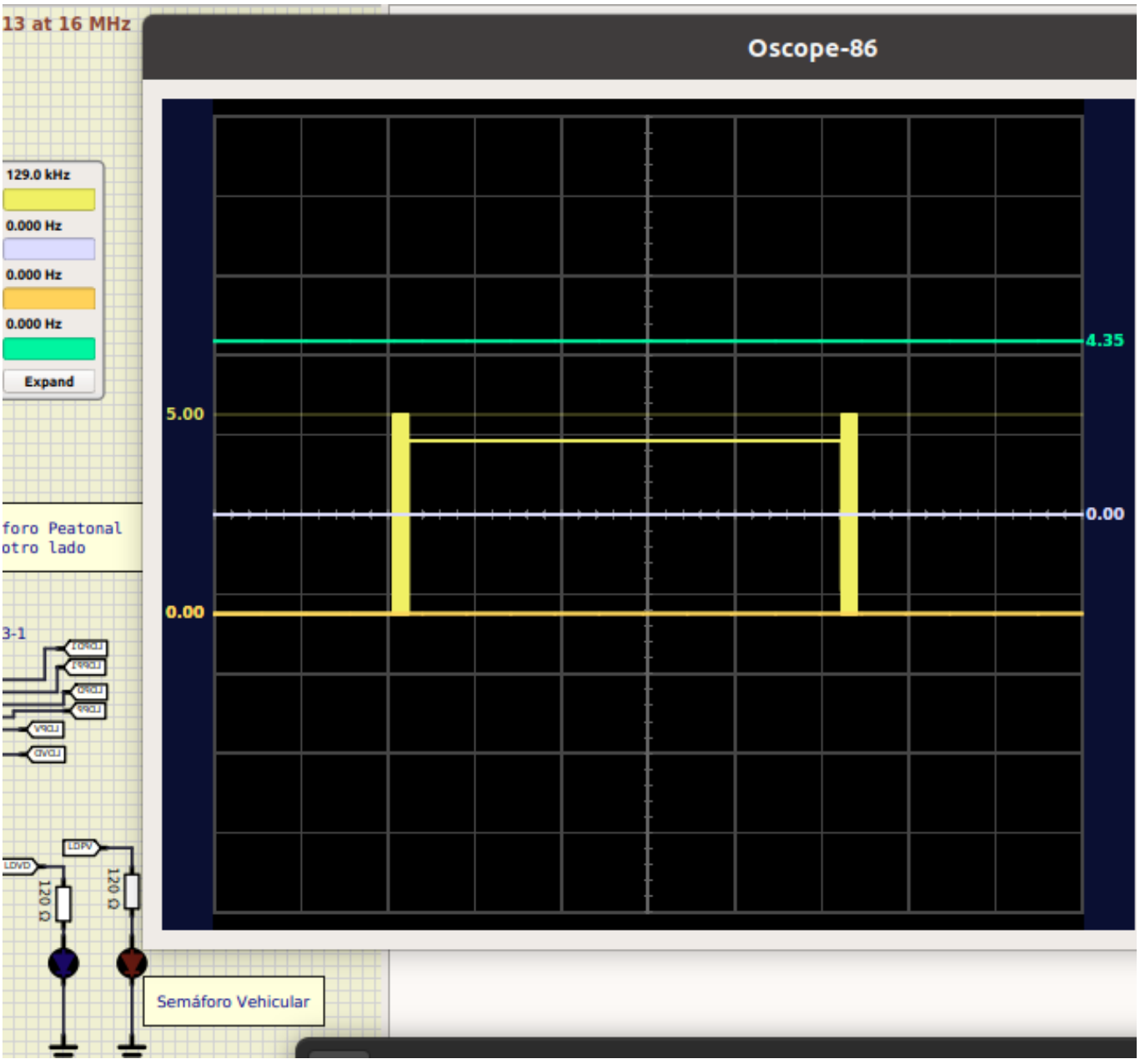


Figura 6: Parpadeo de luz verde de semáforo de carros (LDPV).

Un mayor detalle del funcionamiento se aprecia en la Fig. 7, en el que se está en el estado PPblink (ver Fig. 4).

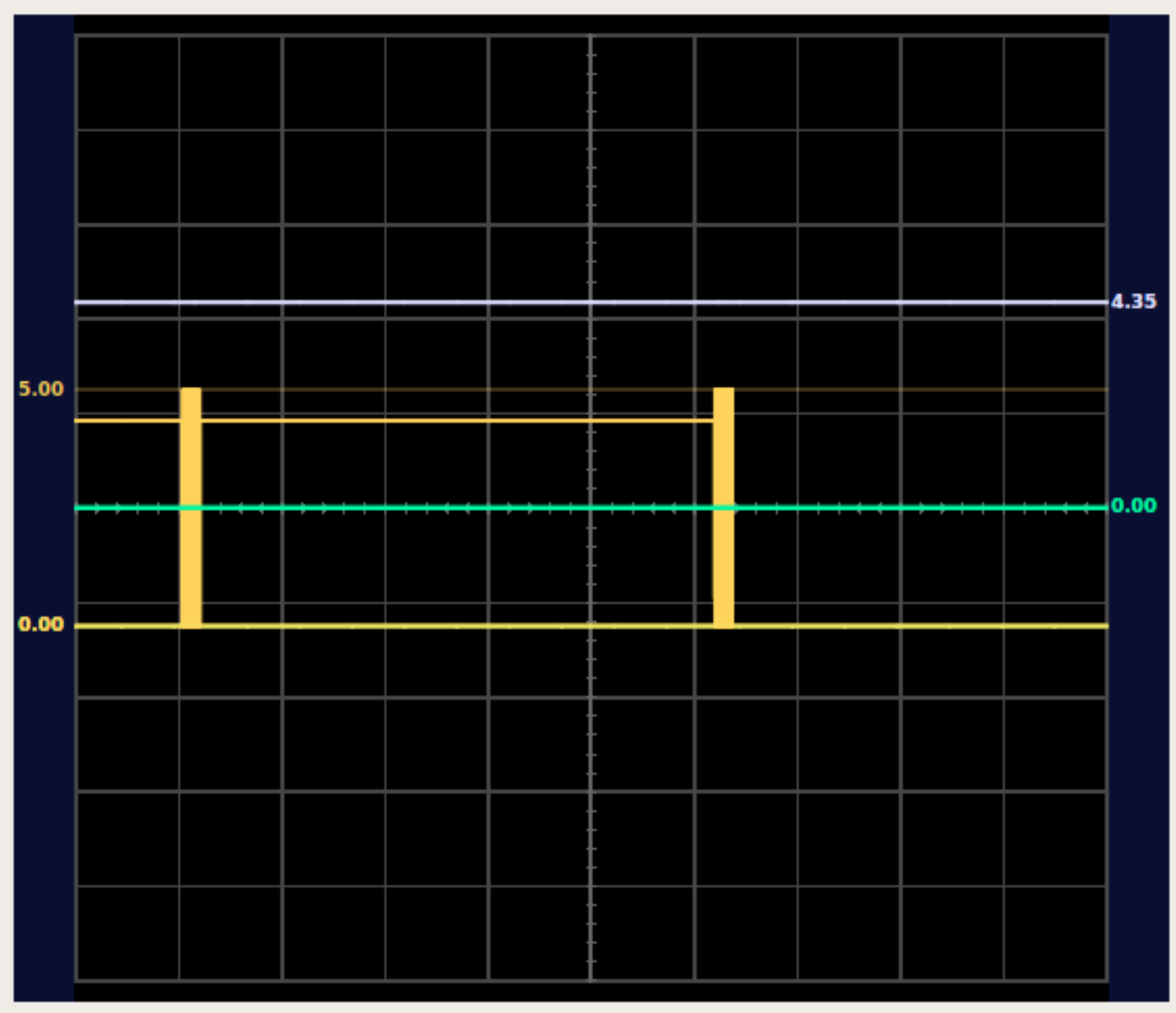


Figura 7: Evidencia de parpadeo de LED de luz verde peatonal (LDPP).

En la Fig. 7 la señal anaranjada conmutando corresponde a la luz LDPP que está parpadeando. La señal verde en 0 V corresponde a la luz roja peatonal (LDPD) que está apagada en este estado. La luz amarilla en 0 V corresponde a la luz verde vehicular (LDPV), también apagada. La luz morada de 4.35 V corresponde a la luz roja vehicular, que en el estado PPblink efectivamente está encendida

3. Conclusiones y recomendaciones

Con la elaboración del presente laboratorio se concluyó que utilizar técnicas de interrupciones para ejecutar ciertas actividades puede resultar más eficiente y rápido que realizarlo todo en el programa principal, ya que esto permite que se ejecuten ciertas acciones en cualquier momento del programa debido a que se activan por hardware. También se concluyó que las máquinas de estados finitas no son útiles solamente en la realimentación de circuitos digitales con memoria, sino también que se pueden usar como paradigma de programación y permiten tener una mejor estructura del funcionamiento deseado. Además, se aprendió como configurar los temporizadores de un microcontrolador y atender interrupciones de por medio de rutinas.

Se tienen las siguientes recomendaciones luego de la realización del laboratorio.

- No empezar el diseño de una vez; es importante primero investigar, ya sea en la hoja de datos o demás, sobre las configuraciones que se deben hacer (registros, rutinas externas a programa principal, etc) antes de empezar con la solución del problema.
- Mejorar el manejo del idioma inglés. Para poder entender con mucha más rapidez el funcionamiento de los registros necesarios, sobre el significado de cada bit de estos y sus posibles configuraciones, un buen nivel de inglés resulta en una utilidad.
- Tener programas más simples que se puedan modificar con ciertas piezas de código del programa principal, para revisar posibles fallos e identificar razones por las cuales el comportamiento del microcontrolador no es el debido.
- A veces, a pesar de haber realizado todo lo posible, pareciera que SimulIDE no logra imitar el comportamiento que el microcontrolador debe tener. En caso de que esto ocurra es importante no frustrarse y cambiar ligeramente algo del diseño para que lo que no funcione vuelva a funcionar.

"The purpose of (scientific) computing is insight, not numbers"
- Richard Hamming.

Referencias

[1] Atmel, *ATtiny2313A/ATtiny4313*. 8246B, 1 ed., 2011.

[2] Multicomp, *Standard Red-emitting LED*. Farnell, 1.1 ed., 2011.

[3] Multicomp, *Green LED - T1 (5mm)*. Farnell, 1.0 ed., 2019. https://www.farnell.com/datasheets/2724776.pdf [online].

[4] Sparkfun, *LED Basic Red 5mm*. Sparkfun Electronics, s.f. https://www.sparkfun.com/products/9590.

[5] Amazon, *Gikfun 6x6x4.3mm TACT Switch Push Button for Arduino*. Amazon.com, Inc, s.f. https://www.sparkfun.com/products/9590.

Apéndices

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

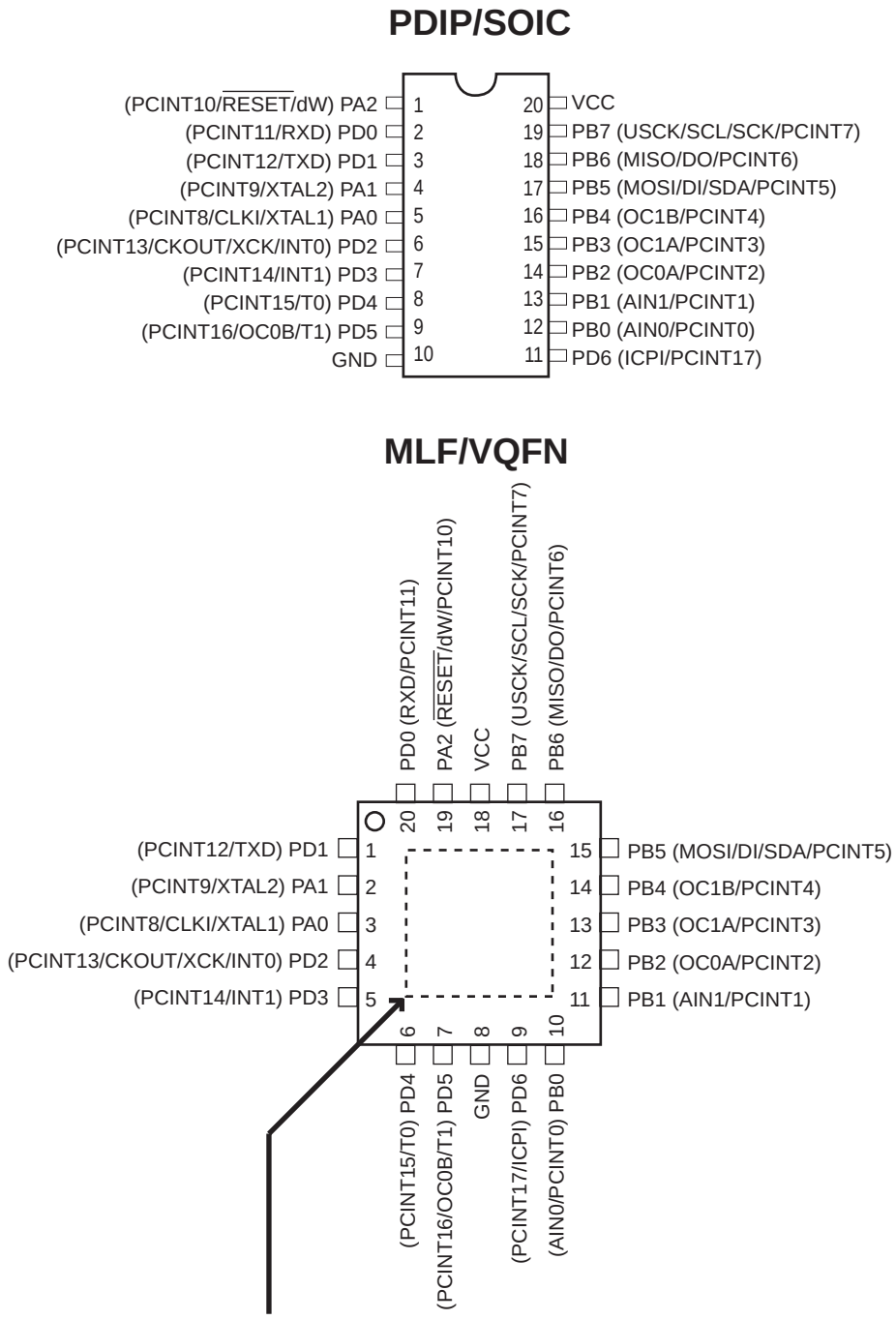
8-bit AVR® Microcontroller with 2/4K Bytes In-System Programmable Flash

ATtiny2313A
ATtiny4313

Rev. 8246B–AVR–09/11

1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313



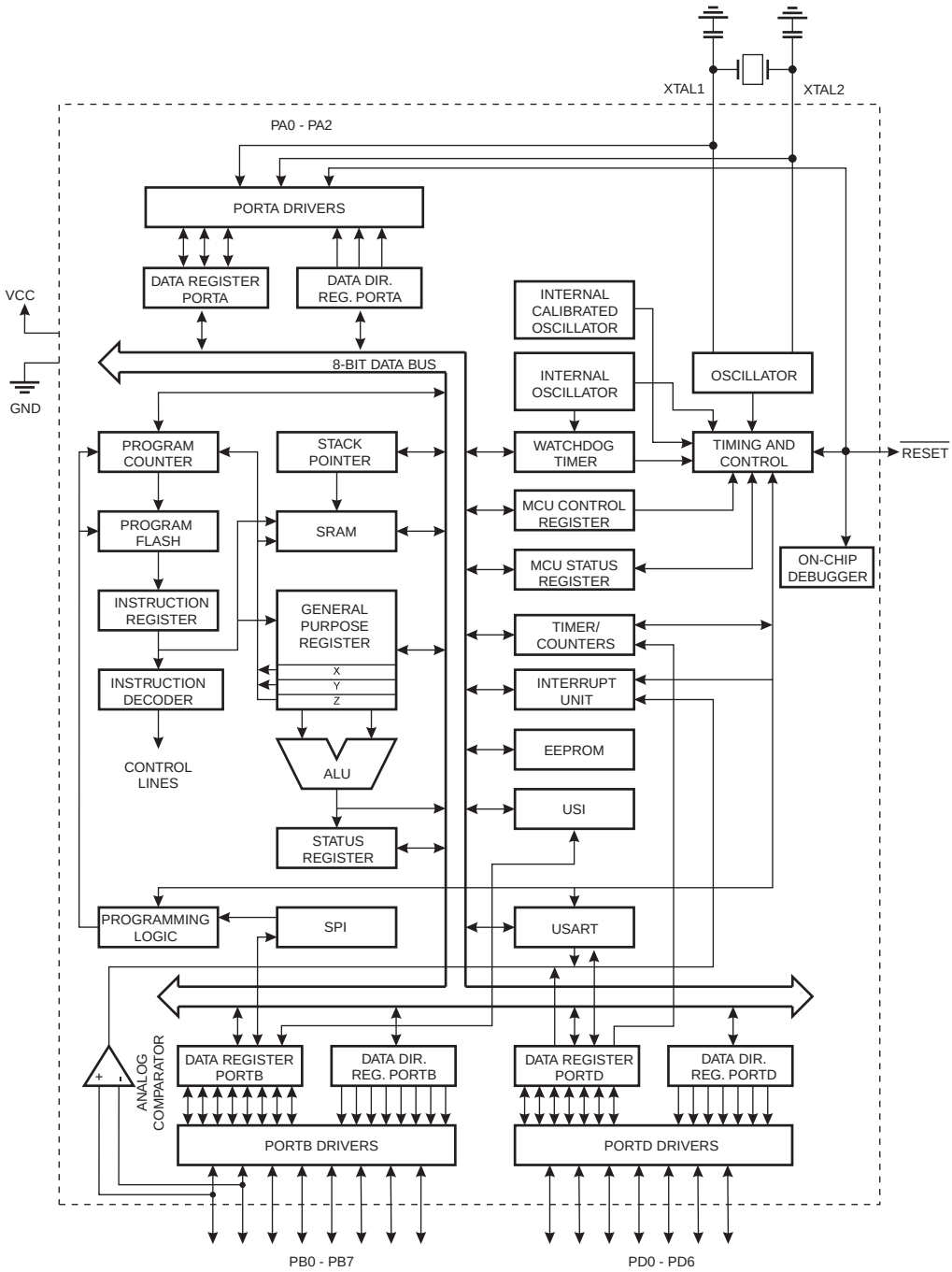
NOTE: Bottom pad should be soldered to ground.

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



Standard LED

Red Emitting Colour

Absolute Maximum Ratings at T_a = 25°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 / °C
Reverse Voltage	5	V
Operating Temperature Range	-25°C to +80°C	
Storage Temperature Range	-40°C to +100°C	
Lead Soldering Temperature (4 mm (0.157) Inches from Body)	260°C for 5 s	

Electrical Optical Characteristics at T_a = 25°C

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Test Condition
Luminous Intensity	I_v		40		mcd	I_f = 20 mA (Note 1)
Viewing Angle	$2\theta_{1/2}$		25		Deg	(Note 2)
Peak Emission Wavelength	λp		640		nm	I_f = 20 mA
Dominant Wavelength	λd		635		nm	I_f = 20 mA (Note 3)
Spectral Line Half-Width	$\Delta\lambda$		25		nm	I_f = 20 mA
Forward Voltage	V_f		2	2.5	V	I_f = 20 mA
Reverse Current	I_R	-	-	100	µA	V_R = 5 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 (λd) is derived from the CIE chromaticity diagram and represents the single wavelength which defines the colour of the device

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 $2\theta^{1/2}$ (°)
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