# Laboratorio de Principios de Mecatrónica

12 de marzo de 2021

---

**Práctica #1**

Microcontroladores

**Grupo:**

L002

**Estudiante:**


- Oceguera Urquiza Juan Manuel
- Jáuregui Tapia Jesús Enrique


**Profesor:**

Benito Granados–Rojas


## Índice

# 1. Introducción

Un microcontrolador es una microcomputadora comprimida diseñada para controlar el funcionamiento de sistemas embebidos. Usualmente, los microcontroladores están compuestos de una serie de elementos básicos: memoria (RAM, ROM, ...), periféricos (ADC, DAC, ...), y un procesador. Así, la gran diferencia entre un microcontrolador y un microprocesador es que el primero tiene "incluidos" todos estos periféricos; por ello, es común que los microcontroladores tengan una menor capacidad de procesamiento. [1]

Frecuentemente, los microcontroladores están conectados a ambientes físicos a través de sensores y actuadores (típicamente son sistemas reactivos). Se componen de hardware (la arquitectura básica ya descrita), y software. Este último se configura alrededor de un set de instrucciones básicas que varía según la familia de controladores a la que pertenece (Microchip, Texas Instruments, EM Microelectronic, ...). Usualmente, además de la posibilidad de programar el microcontrolador a través de este set de instrucciones, también se ofrece la posibilidad de programarlos a través de lenguajes más altos como C.

Existen distintos tipos de arquitecturas para las instrucciones básicas: Complex Instruction Set Computer (CISC), y Reduced Instruction Set Computer (RISC). Las primeras (usualmente asociadas a una arquitectura Von Neumann) ofrecen un amplio y detallado manejo de instrucciones para simplificar los programas; el problema es que la existencia de funciones complejas provoca retardos. Por su parte, la arquitectura RISC (usualmente asociada a la arquitectura Harvard) se distingue por su reducido catálogo de instrucciones, pero mayor eficiencia en la ejecución de ellas. [2]

Se pretende utilizar el lenguaje de programación de Arduino y el set de instrucciones básicas del ATmega2560 para conocer las herramientas y ambientes más comunes para el desarrollo de sistemas mecatrónicos basados en microcontroladores. Después, realizar una comparativa entre las implementaciones en código de bajo y alto nivel, destacando las ventajas y pertinencia de cada una de las estrategias de programación. Además, se pretende realizar distintos ejercicios para identificar los elementos básicos de la arquitectura de un microcontrolador digital, así como los módulos que componen a la tarjeta de desarrollo Arduino y sus interconexiones básicas.

# 2. Experimentos y Simulaciones

Todas las simulaciones de esta práctica fueron realizadas en Tinkercad.

## 2.1. Blink (encendido y apagado de un diodo LED)

En la implementación de este blink se definieron los pines 12 y 13 como salidas para el LED y el botón, respectivamente. Mientras el botón no esté presionado, el

LED parpadea a una frecuencia de 1 Hz; es decir, cada segundo. Al presionar el botón la frecuencia disminuye a 0.5 Hz, o equivalentemente, el periodo aumenta al doble (2 segundos).

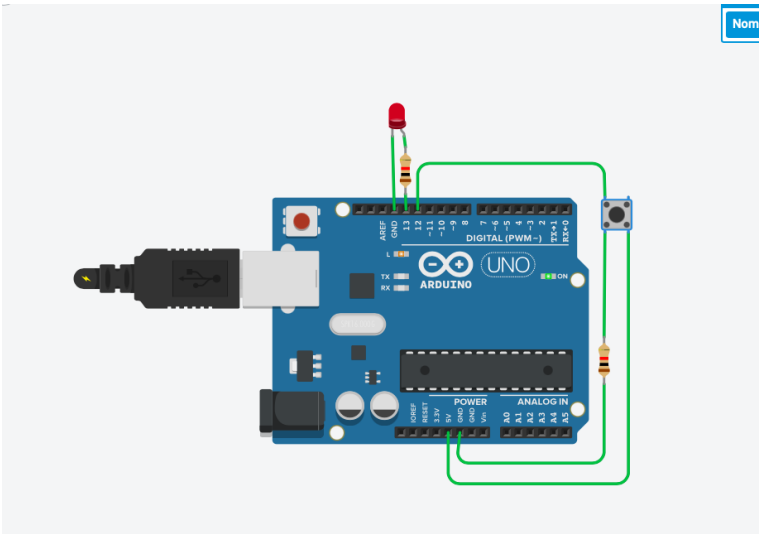


Figura 1: Configuración del Blink con botón.

## 2.2. Blink ASM

Para la implementación del encendido y apagado de un LED en lenguaje ensamblador se utilizó una estructura básica con retrasos predispuestos para que el LED parpadeara a frecuencia de 1 Hz. Para agregar el comportamiento cuando se presiona el botón se utilizó la función 'SBRS' con la que fue posible saltar a una subrutina en caso de que el botón no fuese presionado. En dicha subrutina simplemente se reasignaron los valores del retardo a que fueran aproximadamente la mitad de los originales (de tal manera que cuando el botón fuera presionado tuviera el doble de periodo - o la mitad de frecuencia - que cuando no lo es).

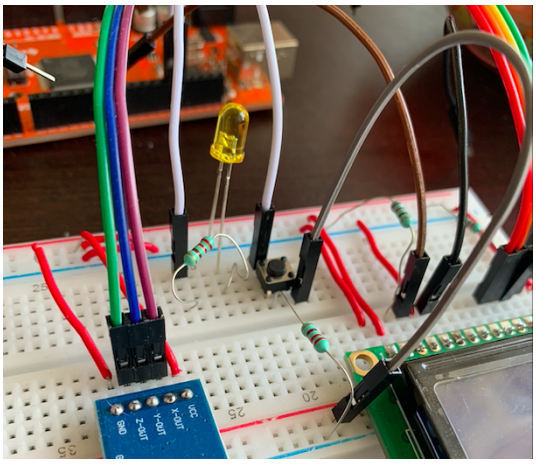

Figura 2: Circuito del Blink con botón en lenguaje ensamblador.

## 2.3. Semáforo digital Arduino

Para la implementación del semáforo, se asignaron los pines 8 a 13 como output de manera que cada uno de ellos represente uno de los 3 colores de cada semáforo.

En esta implementación, se definieron 4 estados que se repiten infinitamente: el primero de ellos es con el LED verde en el semáforo 1 mientras el semáforo 2 tiene encendido el LED rojo; este primer estado cambia después de 5 segundos. Una vez transcurrido el tiempo, el LED amarillo del semáforo 1 se enciende y se apaga el LED verde durante 1 segundo. Posteriormente, se activa el LED rojo en el semáforo 1 y el LED verde en el semáforo 2 durante 5 segundos. Finalmente, en el estado de transición esta vez se enciende el LED amarillo del semáforo 2 (apagando el verde) y dura 1 segundo. Una vez transcurrido este periodo, se reinicia el ciclo.

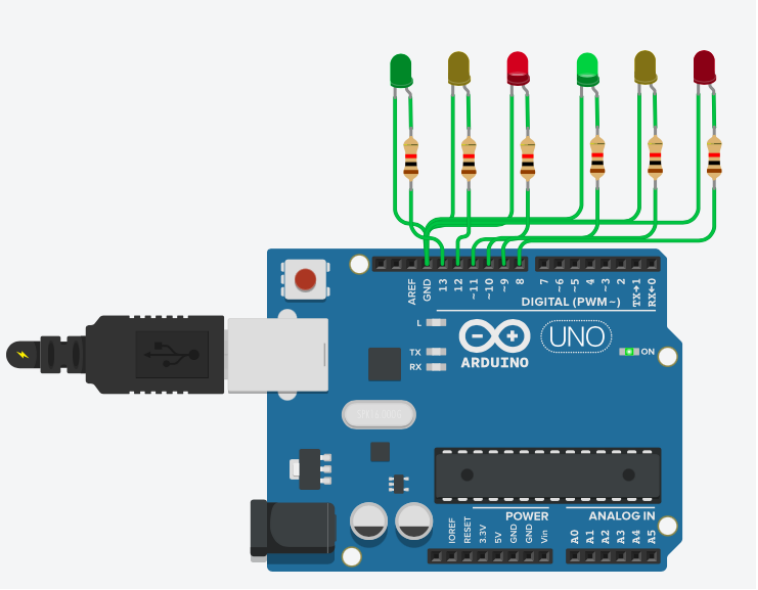


Figura 3: Configuración inicial del semáforo.

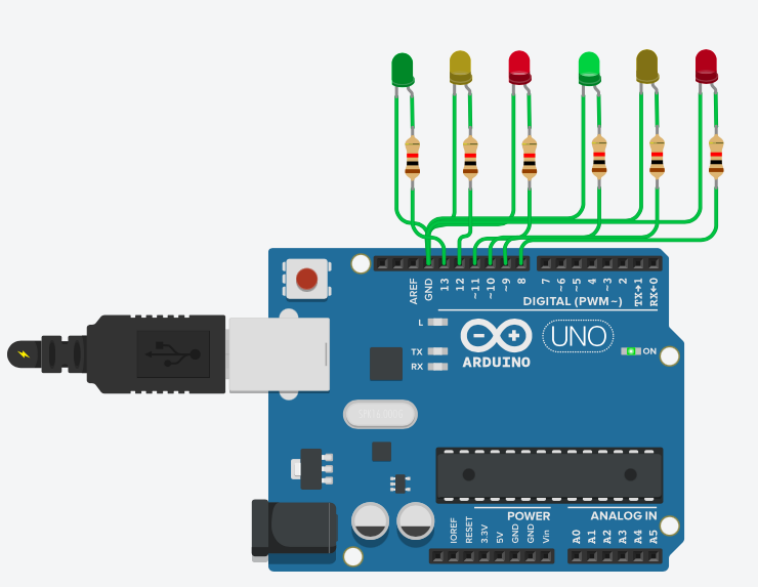


Figura 4: Cambio de estado.

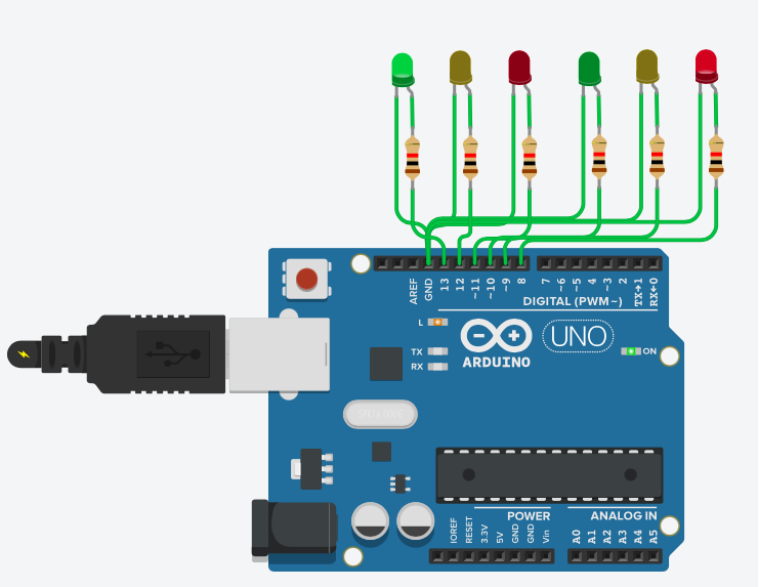


Figura 5: Segunda configuración del semáforo.

# 3. Conclusiones

Es evidente que los microcontroladores son extremadamente útiles para el procesamiento y análisis de señales. En concreto, en la práctica nos dimos cuenta de la facilidad de configurar la tarjeta Arduino para manejar una secuencia de instrucciones en la que liberara energía en ciertos momentos (para encender los LED's).

Además, nos familiarizamos con el uso de lenguaje Ensamblador para emular las tareas que se pueden realizar con el lenguaje Arduino. Si bien en esta práctica no es perceptible la reducción de tiempo y recursos utilizados, es de esperar que utilizar Ensamblador sea más eficiente.

Por último, a través del Blink pudimos conectar la tarjeta Arduino a un elemento externo que determinó su comportamiento (el botón). La posibilidad de ejecutar un set de instrucciones u otro dado un estado externo - en este caso el estado del botón - es de suma importancia para el funcionamiento básico de los microprocesadores: dadas condiciones externas actuar de distintas maneras.

# 4. Enlaces externos

`https://github.com/JuanOceguera/PrincipiosDeMecatronica`.

`https://github.com/Jesus669/PrincipiosMecatronica`

# Referencias

[1] Granados, B. (2021). Principios de Mecatrónica. Tema: *Microcontroladores*. Instituto Tecnológico Autónomo de México. Ciudad de México. 22 de enero del 2020.

[2] Romero, J. G. (2021). Principios de Mecatrónica. Tema: *Sistemas embebidos*. Instituto Tecnológico Autónomo de México. Ciudad de México. 13 de enero del 2020.