Universidad Tecnológica de Panamá

Centro Regional De Veraguas

Facultad de Ingeniería de Sistemas Computacionales

Curso: Circuitos Lógicos

Profesor: Abel Rodríguez

Informe de Laboratorio 5

Estudiantes

Priscila Ortega, Elbin Puga, Arland Barrera, Steven Espinosa

2024

Contenido

Lista de figuras

Marco Teórico

Arduino

Arduino es una plataforma de hardware libre basada en una sencilla placa con un microcontrolador que se puede programar mediante el lenguaje de programación Arduino, una variante simplificada de C/C++. Es ideal para prototipos electrónicos y proyectos educativos, ya que facilita la integración de componentes electrónicos como sensores, actuadores y dispositivos de comunicación. Las placas Arduino pueden interactuar con el entorno a través de pines de entrada y salida digitales o analógicos, que permiten recibir señales y controlar dispositivos conectados.

En este contexto, se utilizan varios periféricos que pueden ser controlados desde el Arduino, como LEDs, buzzers (activos y pasivos), y otros elementos electrónicos. Su fácil manejo y flexibilidad convierten a Arduino en una herramienta potente tanto para principiantes como para ingenieros avanzados que desarrollan prototipos o aplicaciones en áreas como domótica, robótica y el Internet de las Cosas (IoT).

Buzzers Activos y Pasivos

Los buzzers son dispositivos piezoeléctricos que generan sonido. Existen dos tipos principales de buzzers utilizados en proyectos de Arduino: los ***activos*** y los ***pasivos***.

- **Buzzer Activo:** Un buzzer activo contiene un oscilador interno que genera el sonido. Solo necesita una señal de encendido/apagado desde el Arduino para funcionar, lo que simplifica su uso, ya que no requiere una señal de frecuencia de entrada. Esto lo hace ideal para aplicaciones donde se necesita simplemente activar una alerta sonora sin generar una melodía específica o controlar la frecuencia de sonido.

- **Buzzer Pasivo:** A diferencia del activo, el buzzer pasivo no tiene oscilador interno, por lo que necesita una señal de frecuencia externa para generar sonido. Este tipo de buzzer ofrece mayor versatilidad, ya que permite generar tonos específicos, melodías o variaciones de frecuencia, lo cual es útil en proyectos

más complejos como la generación de alarmas con variación de tono o la reproducción de pequeñas melodías. Para utilizar un buzzer pasivo con Arduino, se utiliza la función *tone()*, que permite generar una señal de frecuencia variable para controlar el sonido.

LEDs (Diodos Emisores de Luz)

Los **LEDs** son dispositivos semiconductores que emiten luz cuando son atravesados por una corriente eléctrica. Son ampliamente utilizados en proyectos de electrónica debido a su eficiencia energética, tamaño compacto, durabilidad y bajo costo. Los LEDs pueden emitir luz en diferentes colores y se usan en diversas aplicaciones como indicadores, displays o iluminación.

Un LED tiene dos terminales: el **ánodo** (positivo) y el **cátodo** (negativo). Para encender un LED, se conecta el ánodo a una fuente de corriente positiva y el cátodo a tierra, asegurándose de que la corriente sea lo suficientemente baja como para no dañar el LED. Debido a la baja corriente que soportan los LEDs, es esencial protegerlos con resistencias de limitación de corriente.

Resistencias de Protección

Las **resistencias** son componentes electrónicos que limitan el paso de la corriente en un circuito. En el caso de los LEDs, una resistencia de protección es crucial para evitar que el LED se queme por exceso de corriente. La cantidad de corriente que puede soportar un LED es limitada, generalmente unos 20 mA, y si se le aplica una corriente mayor, el componente puede dañarse.

Para calcular el valor de la resistencia necesaria, se utiliza la **Ley de Ohm:**

$$R = \frac{V_f - V_{LED}}{I_{LED}}$$

Donde:

- R es el valor de la resistencia en Ω (ohmios).
- V_f es el voltaje de la fuente (por ejemplo, 5V en el caso de Arduino).
- V_{LED} es el voltaje de funcionamiento del LED (generalmente entre 1.8 y 3.3V, dependiendo del color).
- I_{LED} es la corriente que debe pasar por el LED (típicamente 20 mA o 0.02 A).

Una resistencia comúnmente usada para proteger LEDs es de **220** Ω, lo cual es adecuado para la mayoría de los proyectos educativos con Arduino.

Sensores y Pines de Entrada y Salida

En proyectos de Arduino, los sensores permiten recoger datos del entorno, como temperatura, luz o distancia. Los pines **digitales** de entrada y salida (I/O) permiten leer señales binarias (alto o bajo, 1 o 0), mientras que los pines **analógicos** permiten leer valores continuos, como la salida de un sensor de luz (LDR) o un potenciómetro.

Los pines PWM (Pulse Width Modulation) son de gran utilidad cuando se necesita simular una señal analógica. Aunque Arduino tiene pines digitales que solo pueden estar en alto (5V) o bajo (0V), mediante PWM se puede generar una señal que simula voltajes intermedios. Este principio es usado, por ejemplo, para controlar la intensidad de un LED o variar el tono de un buzzer pasivo.

Función de Tono y Control de Sonido

La función ***tone()*** de Arduino es fundamental cuando se trabaja con buzzers pasivos. Permite generar señales de frecuencia variable que controlan el sonido emitido por el buzzer. El uso de esta función es esencial para desarrollar aplicaciones de alerta sonora o para generar melodías.

Introducción

El presente informe de laboratorio tiene como objetivo explorar el uso de la plataforma Arduino para la construcción y programación de circuitos electrónicos. En particular, se desarrollan una serie de ejercicios prácticos que incluyen el uso de buzzer piezoeléctrico, LEDs y resistencias, aplicando el conocimiento de programación y electrónica básica. El propósito es fomentar la comprensión del funcionamiento de los circuitos lógicos y cómo interactúan con el entorno a través de señales digitales y analógicas.

Los experimentos abarcan desde el encendido de LEDs hasta la generación de tonos con un buzzer, permitiendo a los estudiantes familiarizarse con la programación de Arduino y el montaje de circuitos sencillos. Se busca que los participantes dominen la manipulación de los componentes, el análisis de circuitos, y la interacción con el software de Arduino para programar comportamientos predefinidos.

Cuerpo de Trabajo

3.1 Objetivos Generales

Manipular adecuadamente la placa de arduino para completar los circuitos propuestos y posteriormente, comprender las conexiones de la placa arduino con el fin de indagar que funcionamiento tiene en la aplicación a los circuitos.

3.2 Objetivos Específicos

- Entender el código de programación de la placa arduino.
- Entender el funcionamiento de los circuitos desarrollados.
- Comprender el funcionamiento del piezo (buzzer) y como impacta al funcionamiento del circuito.

3.3 Materiales

Figura 3.1: Resistor de 10 $k\Omega$

Figura 3.2: Buzzer activo

Desarrollo

4.1 Circuitos Propuestos

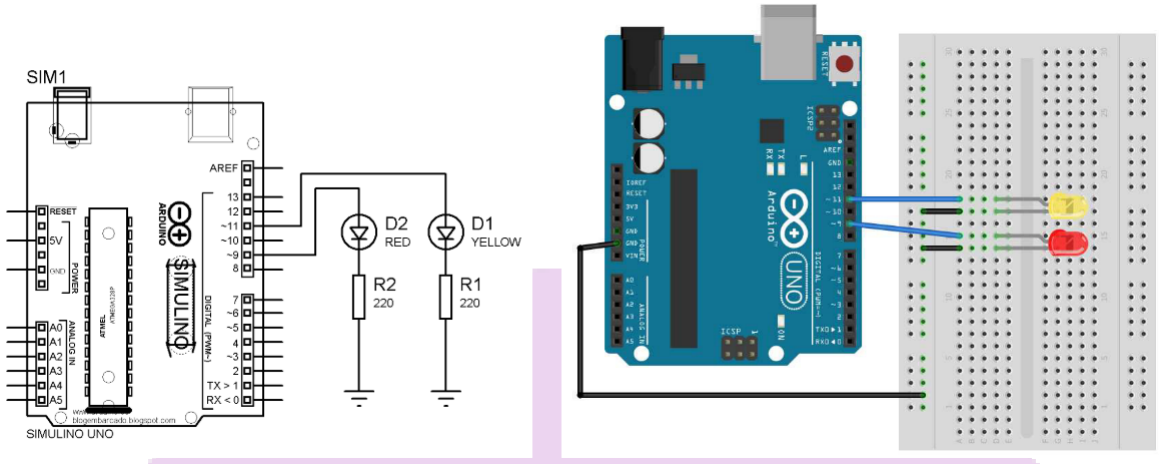

Figura 4.1: Circuito propuesto 9

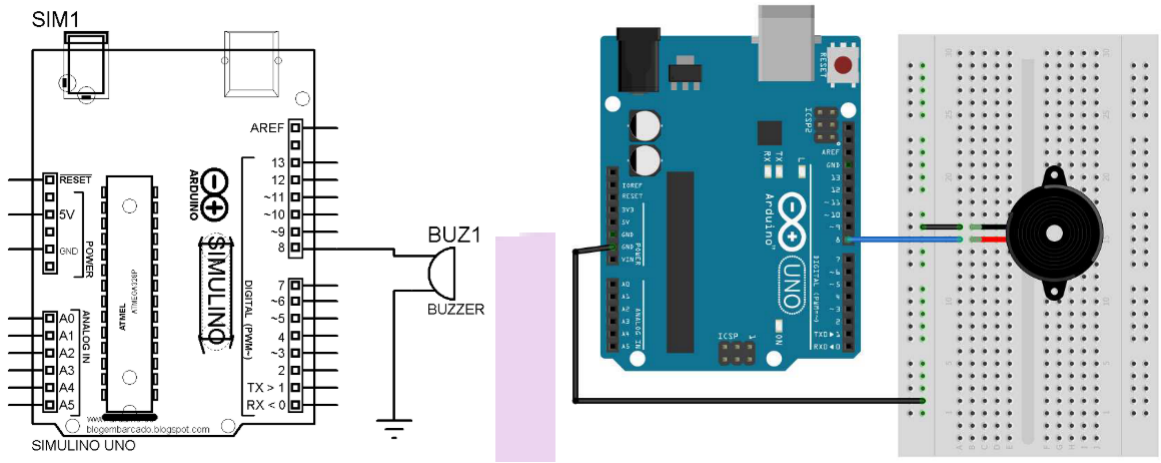

Figura 4.2: Circuito propuesto 10

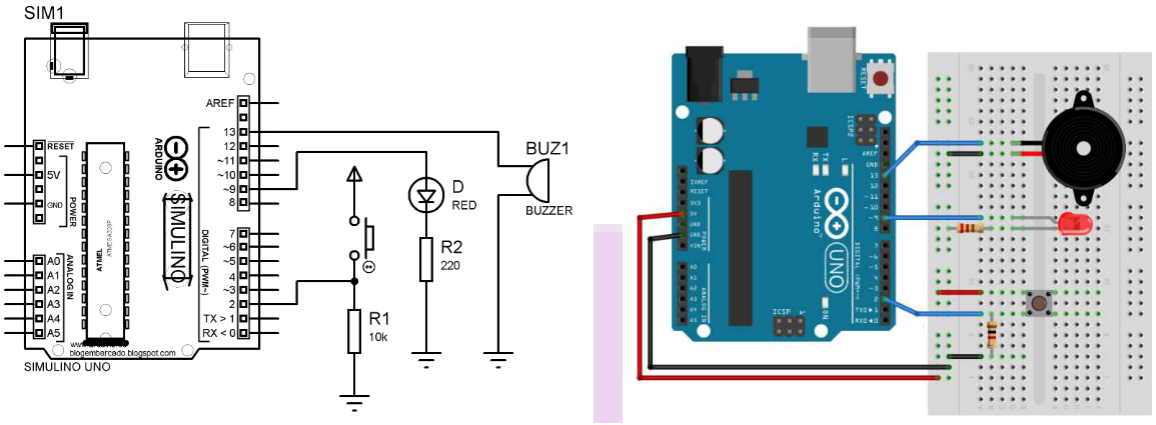

Figura 4.3: Circuito propuesto 11

4.2 Circuitos Desarrollados

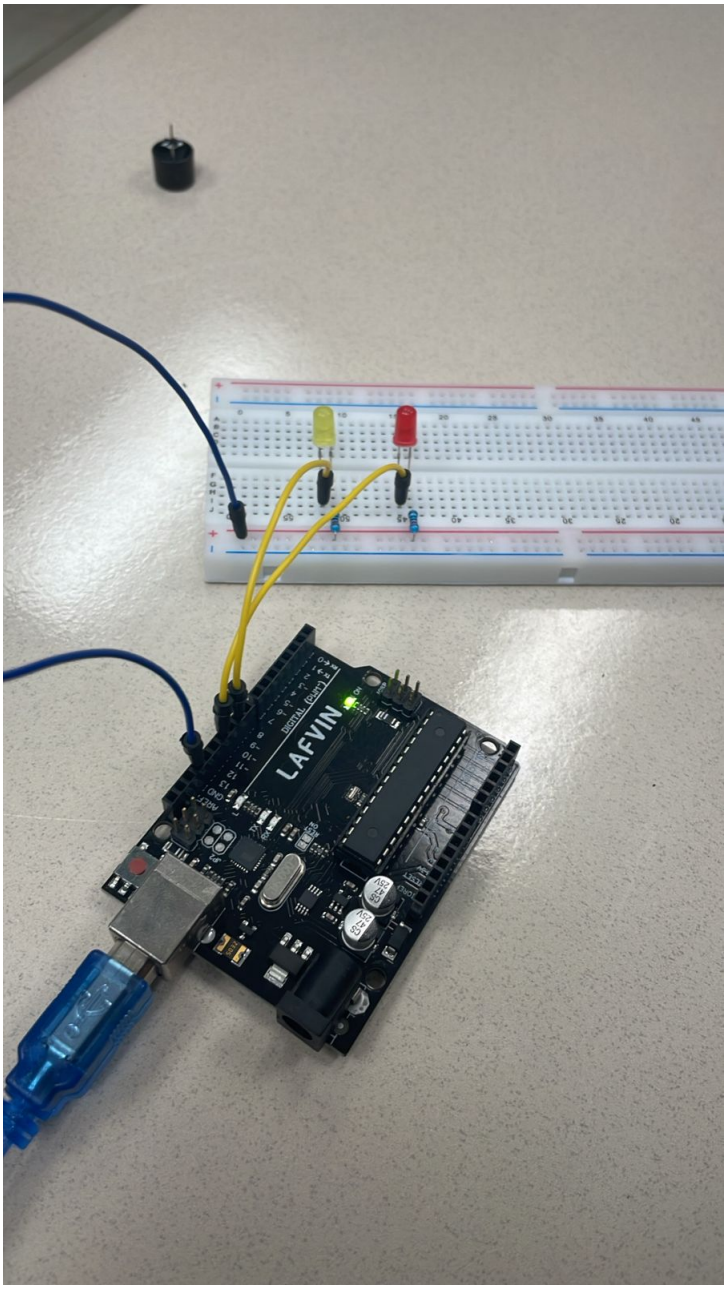

Figura 4.4: Circuito 9 armado

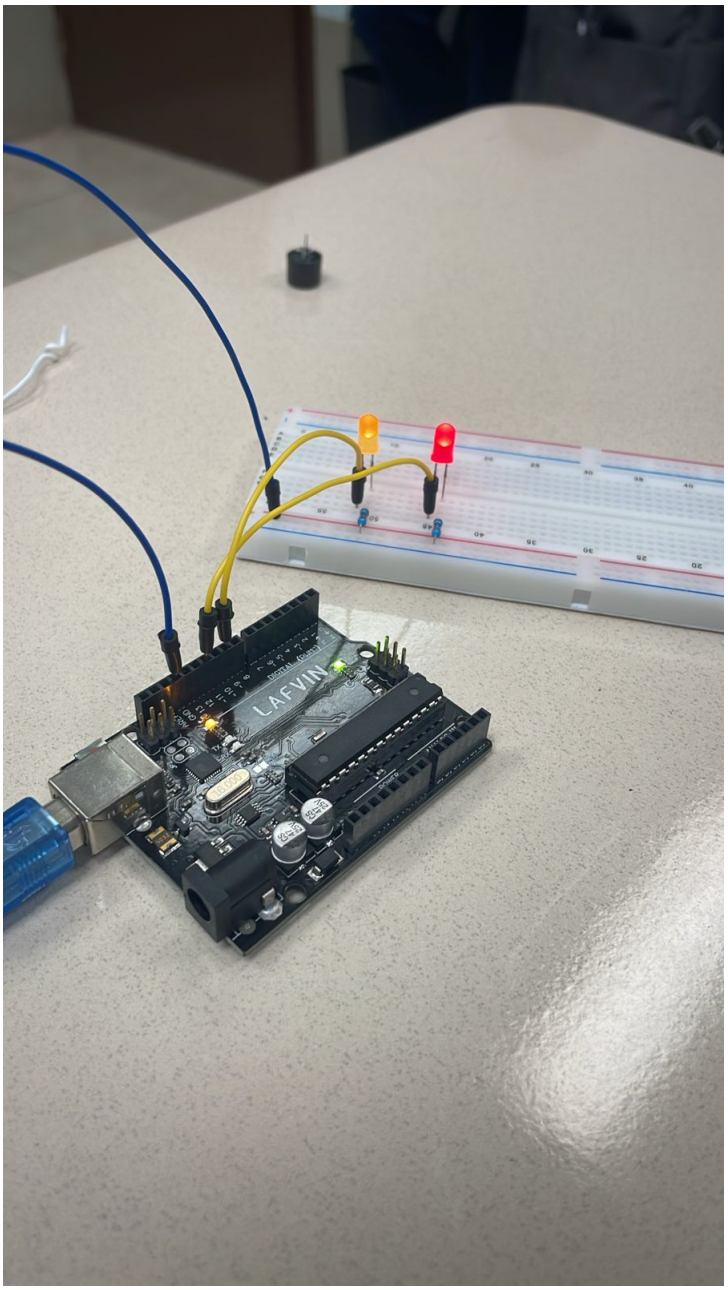

Figura 4.5: Circuito 9 ejecución

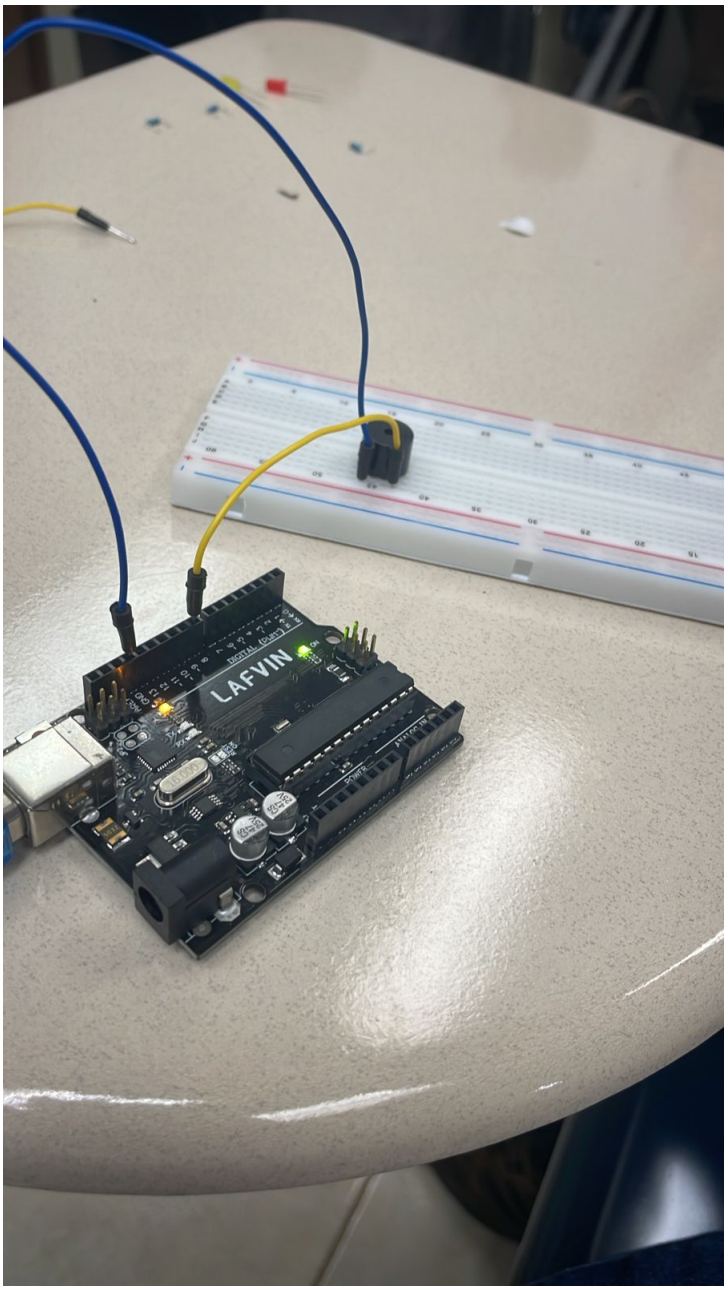

Figura 4.6: Circuito 10 armado

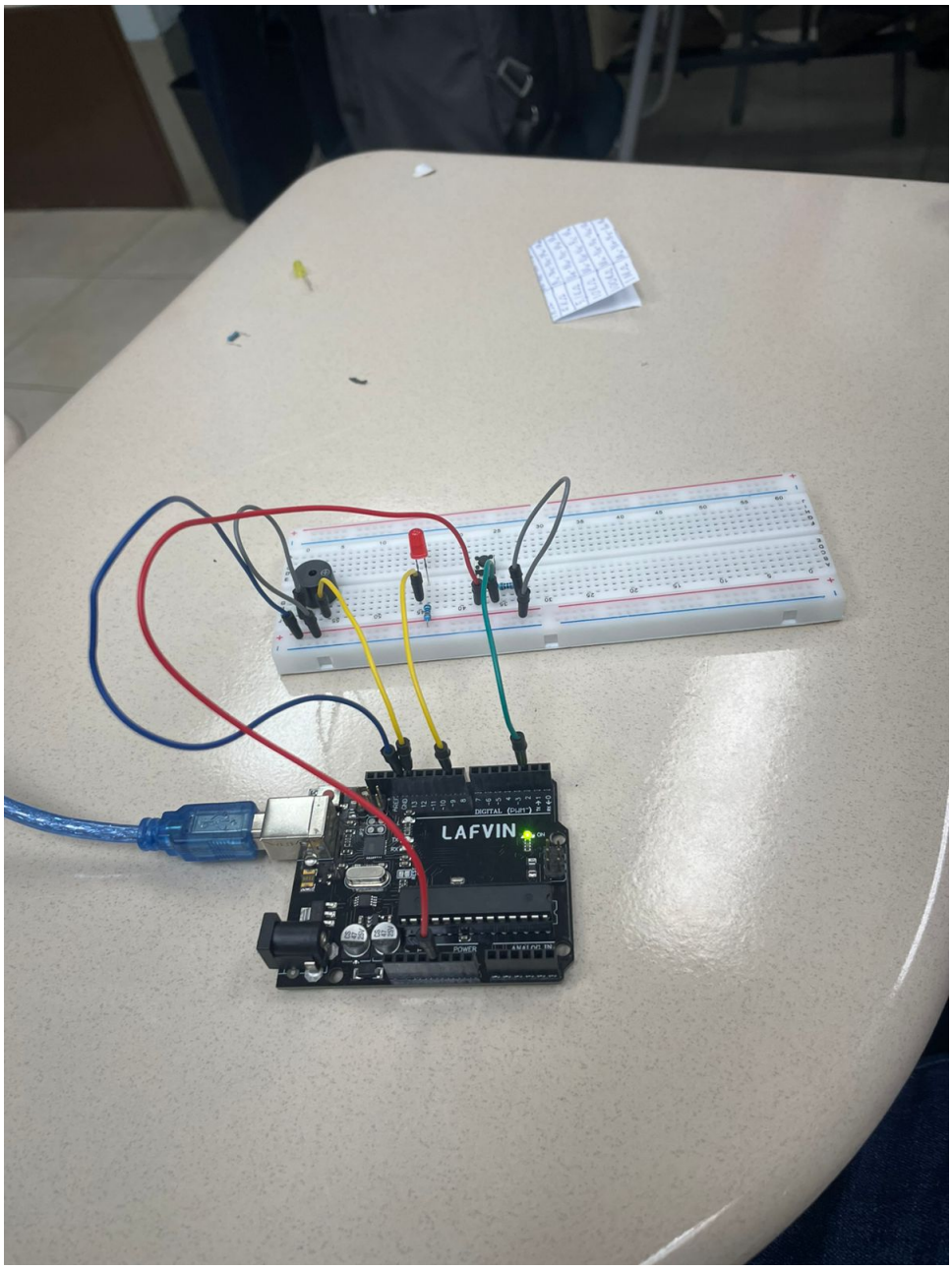

Figura 4.7: Circuito 11 armado

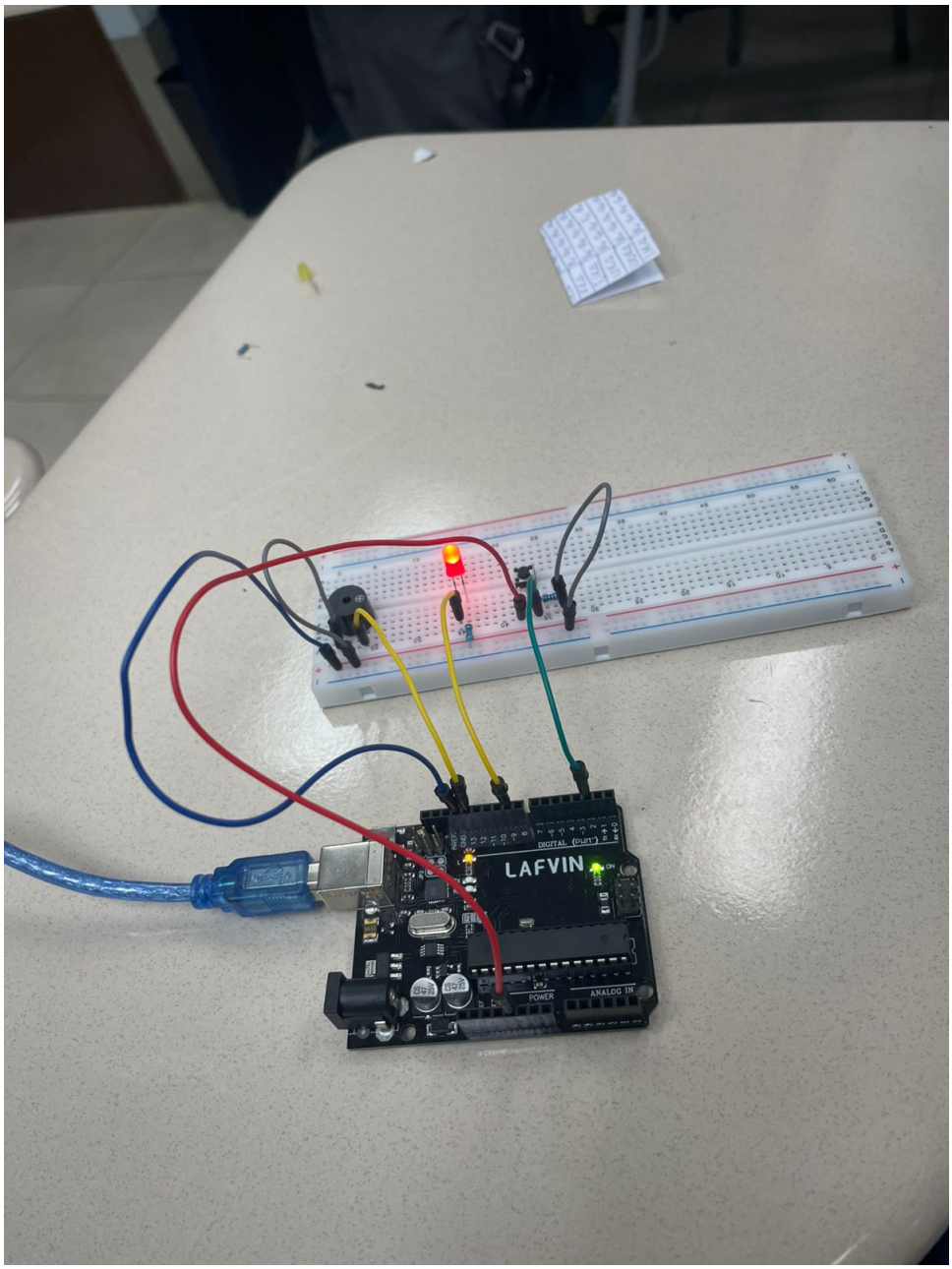

Figura 4.8: Circuito 11 ejecución

Cuestionario

5.1 Circuito 9

¿Cómo modificarías el sketch para poder visualizar también por el serial los dos números ale torios generados en cada loop?

Código 9 Listings

Cambios principales:

1. *Serial.begin(9600)*: Inicializa la comunicación serial a 9600 baudios.

2. *Serial.print()* y *Serial.println()*: Muestra los valores aleatorios generados para los LEDs en el monitor serial.

5.2 Circuito 10

¿Cómo harías para que el piezo tocara alguna melodía?

Código 10 Listings

Cambios principales:

1. Arrays *melody[]* y *noteDurations[]*: El array *melody[]* contiene las frecuencias de las notas (en Hertz), y el array *noteDurations[]* especifica la duración relativa de cada nota.

2. *noteDuration = 1000 / noteDurations[i]*: Calcula la duración real de cada nota en milisegundos.

3. *tone(buzzer, melody[i], noteDuration)*: Reproduce la nota correspondiente del array *melody[]* con la duración calculada.

4. *delay(noteDuration * 1.30)*: Añade una ligera pausa entre las notas para que suene de manera más natural.

5.3 Circuito 11

¿Cómo harías para que en vez de 10 ciclos fuesen 15 ciclos y cada ciclo durase 20 ms menos?

Código 11 Listings

Cambios principales:

1. Ciclos en el *for*: Cambiar el límite de *i <= 10* a *i <= 15* para hacer 15 ciclos.
2. Duración de los ciclos: Reducir los *delay(100)* a *delay(80)* para que cada ciclo sea 20 ms más corto ($100ms - 20ms = 80ms$).

Conclusiones

A lo largo de la práctica de laboratorio, se logró un dominio fundamental en el uso de Arduino para controlar dispositivos electrónicos, así como en la programación de su microcontrolador. Se pudo evidenciar que la plataforma es accesible tanto para principiantes como para proyectos más avanzados, debido a su flexibilidad y a la vasta cantidad de recursos disponibles.

El uso de circuitos sencillos con LEDs y buzzers permitió a los estudiantes comprender el funcionamiento básico de los componentes electrónicos, así como su programación. Asimismo, la interacción con la comunicación serie facilitó el monitoreo de datos en tiempo real, una habilidad crucial para el desarrollo de proyectos electrónicos más complejos.

Finalmente, se destaca la importancia de la correcta interpretación de los esquemas eléctricos y la adecuada disposición de los componentes en un protoboard, lo que facilita la comprensión teórica y la práctica de conceptos de electrónica y programación.

Referencias

Arduino R3. (2024). *Arduino product reference manual. https://docs.arduino.cc/

OIKOS MATEMATIKÓN PRÁCTICAS DE ARDUINO. Proporcionado por el profesor Abel Rodríguez

Anexos

8.1 Datos de los materiales usados

1. **Circuito Integrado (IC) de Arduino**
   - **Modelo:** LAFVIN Arduino R3
   - **Microcontrolador:** ATmega328P (similar al Arduino UNO)
   - **Arquitectura:** 8 bits
   - **Memoria Flash:** 32 KB
   - **RAM:** 2 KB
   - **EEPROM:** 1 KB
   - **Pines Digitales I/O:** 14
   - **Pines Analógicos I/O:** 6
   - **PWM:** 6
   - **Timers:** 3
   - **UARTs:** 1
   - **SPI:** 1
   - **12C:** 1
2. **LEDs (Diodos Emisores de Luz)**
   - **Tipo:** LED estándar de 5 mm
   - **Color:** Rojo y amarillo
   - **Tensión de Funcionamiento (VF):** 2-3.5V
   - **Corriente de Funcionamiento:** 20 mA
3. **Jumpers (Cables de Conexión)**
   - **Tipo:** Cables de conexión macho-macho, macho-hembra, hembra-hembra

- **Longitud:** 15 cm a 30 cm

4. **Pulsadores (Botones)**
   - **Tipo:** Pulsador momentáneo
   - **Número de Pines:** 4
   - **Tensión Máxima:** 12 V
   - **Corriente Máxima:** 0.5 A
   - **Dimensiones:** Aproximadamente 6x6 mm
5. **Protoboard (Breadboard)**
   - **Tipo:** Protoboard estándar
   - **Dimensiones:** Comúnmente 830 puntos
6. **Resistor de 220 Ω (Ohmios) a 1/4 W**
   - **Valor:** 220 Ω
   - **Tolerancia:** ±1
   - **Potencia Nominal:** 12 V
7. **Resistor de 10 $k\Omega$ (Ohmios) a 1/4 W**
   - **Valor:** 10 $k\Omega$
   - **Tolerancia:** ±1
   - **Potencia Nominal:** 12 V
8. **Piezo (Buzzer) Activo**
   - **Voltaje Nomianl:** 5 V
   - **Voltaje de funcionamiento:** 4 - 8 V
   - **Corriente nominal máxima:** <=32 mA
   - **Frecuencia resonante:** 2300 ±300 Hz
   - **Temperatura de funcionamiento:** -20°C to 45°C
   - **Dimensiones (sin pines)**
     - **Altura:** 9.16 mm (0.36")
     - **Diámetro:** 11.78 mm (0.46")
     - **Peso:** 1.6 g (0.057 oz)

8.2 Alcance de la aplicación de este laboratorio a futuro

El laboratorio de Arduino tiene un amplio alcance para aplicaciones futuras, especialmente en el campo de la automatización, el control de sistemas y la enseñanza de electrónica básica. Los conocimientos adquiridos pueden aplicarse en áreas como la robótica, el internet de las cosas (IoT), y el desarrollo de sistemas embebidos.

A nivel educativo, este tipo de prácticas fomenta el aprendizaje activo y permite a los estudiantes aplicar sus conocimientos en la creación de prototipos reales, lo que a su vez refuerza la comprensión de los principios teóricos de la electrónica. En el ámbito profesional, las aplicaciones prácticas de Arduino permiten diseñar sistemas de control eficientes y económicos, abriendo puertas a innovaciones tecnológicas tanto en el ámbito industrial como en el doméstico.