

Arduino

Arduino es una plataforma de hardware y software de código abierto que permite a los usuarios crear proyectos interactivos mediante la programación de una pequeña placa electrónica. En el contexto educativo, Arduino se ha convertido en una herramienta fundamental para enseñar robótica, programación y diseño de circuitos, especialmente en proyectos STEAM (ciencia, tecnología, ingeniería, arte y matemáticas). Su facilidad de uso y la gran variedad de sensores y actuadores disponibles la hacen ideal para llevar a cabo proyectos de robótica y arte en el aula, donde los estudiantes pueden crear artefactos interactivos, instalaciones de luz, y experimentos de sonido.

1 La placa Arduino

La placa Arduino es un microcontrolador que puede programarse para interactuar con distintos sensores y actuadores. Esto se realiza mediante el software de Arduino, donde los usuarios escriben código (en lenguaje C++) que luego se carga en la placa para controlar dispositivos conectados a ella. En términos prácticos, Arduino permite a los estudiantes dar instrucciones a los componentes del circuito para que realicen acciones, respondan a estímulos del entorno y generen movimientos o efectos visuales y sonoros.

La gran ventaja de Arduino en proyectos de robótica y arte es su capacidad para conectar componentes diversos, como LEDs, servomotores, sensores de distancia, pulsadores, y mucho más, en una misma placa. Esto hace posible que los estudiantes diseñen proyectos que respondan al entorno, realicen movimientos o emitan sonidos, lo que permite combinar la tecnología con la creatividad de una manera efectiva y accesible.

2 El Entorno de desarrollo integrado para Arduino (Arduino IDE)

Arduino utiliza un entorno de desarrollo integrado (IDE) que facilita la programación y la carga de instrucciones a la placa de Arduino desde cualquier computadora. Este entorno permite escribir, editar, compilar y enviar el código directamente a la placa, y se caracteriza por su simplicidad y accesibilidad, lo que lo hace ideal para principiantes, incluidos los estudiantes de educación primaria y secundaria.

El Arduino IDE (Integrated Development Environment) es el programa principal que se utiliza para escribir y cargar código en las placas de Arduino. Es un software gratuito y de código abierto que puede instalarse en la mayoría de los sistemas operativos, como Windows, macOS y Linux.

El IDE tiene una interfaz clara, con un editor de texto para escribir código y una serie de botones para compilar y cargar el programa en la placa. Incluye también un monitor serial, que permite ver datos en tiempo real desde la placa, lo que es útil para proyectos que requieran monitorear sensores o interactuar con el entorno.

2.1 Estructura Básica del Código en Arduino

El código en Arduino se escribe en un lenguaje de programación basado en C++, pero con una estructura simplificada para facilitar el aprendizaje. Los programas o "sketches" de Arduino siguen siempre una estructura básica que incluye dos funciones esenciales:

- **void setup():** Esta función contiene el código que se ejecuta solo una vez al iniciar la placa.

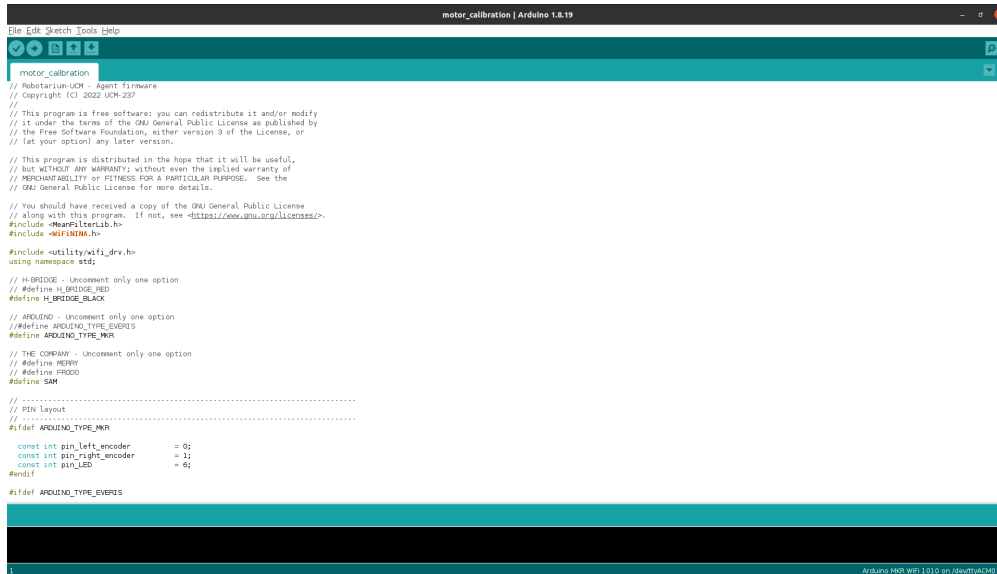


Figure 1: Arduino IDE

Aquí se configuran los pines de entrada y salida, se inicializan sensores, actuadores y se establecen configuraciones iniciales.

- **void loop():** Esta función contiene el código que se ejecuta en bucle, es decir, repitiéndose continuamente mientras la placa esté encendida. Aquí se definen las acciones que queremos que la placa realice de forma repetitiva, como leer datos de sensores, activar motores o mostrar información.

2.2 Programación con Arduino

El Arduino IDE permite realizar distintos niveles de programación, que pueden adaptarse a la edad y nivel de conocimientos de los estudiantes. Además del lenguaje de Arduino, también existen plataformas que permiten programar la placa de forma visual, como Tinkercad.

Arduino cuenta con una amplia colección de librerías (colecciones de código preescrito) que facilitan el uso de componentes y módulos avanzados. Las librerías permiten simplificar el código, especialmente en proyectos que integren sensores o actuadores más complejos, como el sensor de ultrasonidos o los servos, reduciendo la programación necesaria para su funcionamiento.

2.3 Monitorización y Depuración

Una de las herramientas más útiles del Arduino IDE es el Monitor Serial, que permite ver en tiempo real los datos que se generan en la placa. Esto es especialmente útil en proyectos que utilizan sensores, ya que permite observar y depurar el funcionamiento del código, identificando errores o analizando la respuesta de los sensores en distintos contextos. Esta función ayuda a que los estudiantes comprendan mejor cómo interactúa su código con el entorno, promoviendo la experimentación y la resolución de problemas.

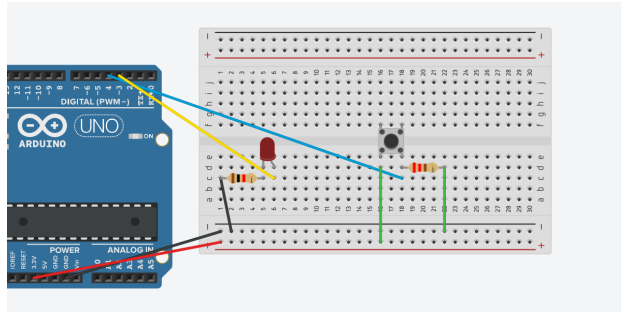


Figure 2: Conexiones LED-Botón

3 Proyectos con Arduino

El objetivo de esta primera práctica es aprender a controlar LEDs mediante un pulsador y experimentar con la lógica básica de programación en Arduino.

Tarea 1: Luces Interactivas con LEDs de Colores y un Pulsador

Vamos a construir un circuito simple que encienda y apague LEDs de colores al presionar un pulsador. Trabajaremos los conceptos de entrada y salida digital, así como el uso básico de resistencias y el código de control.

Componentes:

- Placa Arduino
- LEDs de colores
- Pulsador
- Resistencias (330Ω para LEDs)
- Placa protoboard
- Cables de conexión

Pasos

1. Conectar los LEDs y el pulsador a la protoboard y a la placa Arduino según el diagrama de conexión, Figura 2
2. Programar en el IDE de Arduino para que el LED encienda al pulsar el botón y se apague al soltarlo.
3. Modificar el programa para hacer que los LEDs parpadeen o se enciendan en secuencia al presionar el botón.

Exploraremos el control analógico en Arduino para ajustar el brillo de un LED RGB y crear combinaciones de color usando un potenciómetro.

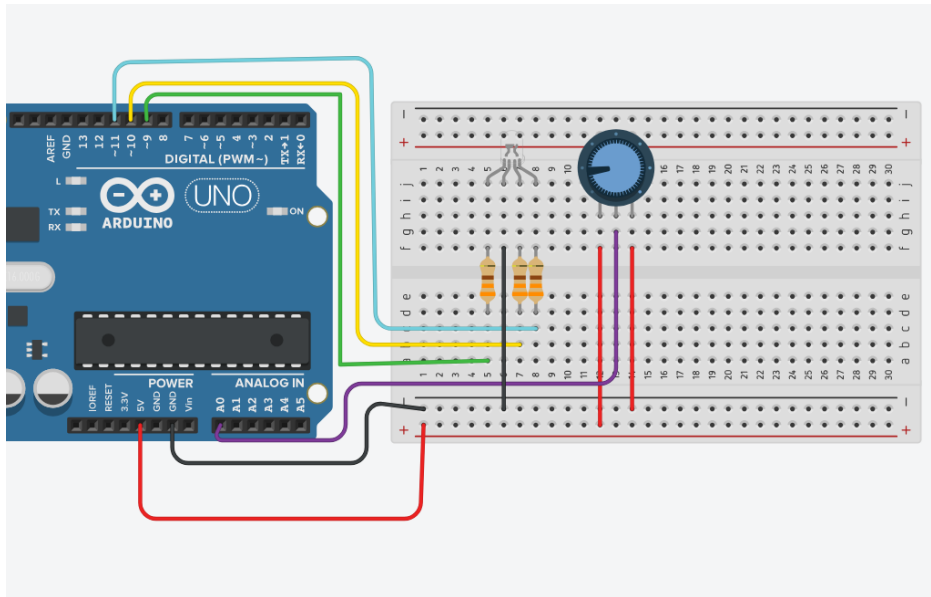


Figure 3: Conexión potenciómetro y LED RGB

Tarea 2: Control de Brillo y Color con LED RGB y Potenciómetro

Cambiaremos los colores de un LED RGB mediante un potenciómetro, experimentando con la creación de efectos de luz y combinaciones de colores.

Componentes:

- Placa Arduino
- LED RGB
- Potenciómetro
- Resistencias (330Ω)
- Placa protoboard
- Cables de conexión

Pasos:

1. Conectar el LED RGB y el potenciómetro en la protoboard, y conectar ambos a la placa Arduino, figura 3.
2. Programar para que el potenciómetro controle el brillo de cada canal de color (rojo, verde y azul) del LED RGB.
3. Probar distintas combinaciones de colores ajustando el potenciómetro y experimentar con la creación de patrones de color.

Podemos utilizar un sensor de distancia para activar un zumbador y crear una instalación interactiva que responda al movimiento.

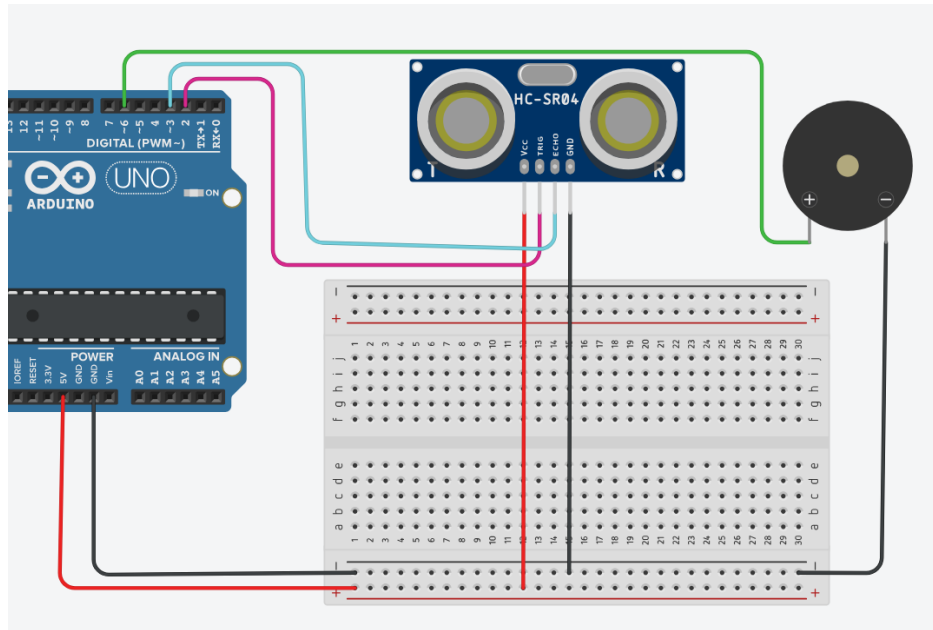


Figure 4: Conexión sensor de ultrasonidos y zumbador

Tarea 3: Escultura Interactiva con Sensor de Ultrasonidos y Zumbador

Construiremos una pequeña "escultura sonora" que reaccione a la presencia de una persona. Cuando alguien se acerque a la escultura, el sensor de ultrasonidos detectará la distancia y activará un sonido en el zumbador.

Componentes:

- Placa Arduino
- Sensor de ultrasonidos
- Zumbador
- Placa protoboard
- Cables de conexión

Pasos:

1. Conectar el sensor de ultrasonidos y el zumbador a la protoboard y al Arduino, figura 4.
2. Programar el Arduino para que lea la distancia del sensor y active el zumbador cuando detecte una distancia menor a un valor específico.
3. Ajustar la sensibilidad y el sonido del zumbador según la distancia para crear un efecto interactivo.

Podríamos también combinar el uso de sensores y actuadores para crear una figura móvil que se mueva en función de la luz ambiente.

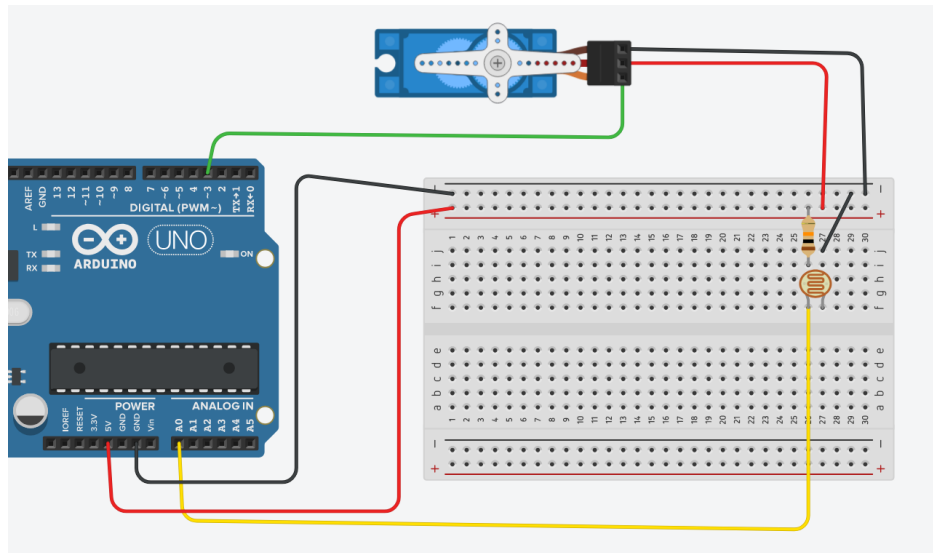


Figure 5: Conexión del servomotor y LDR

Tarea 4: Figura Móvil con Mini Servo Controlado por LDR (Sensor de Luz)

En este proyecto usaremos un LDR (sensor de luz) para controlar un mini servomotor, creando una figura que se mueva o ajuste su posición en respuesta a los cambios de luz. Esta actividad permite integrar conceptos de arte y robótica para desarrollar una instalación que reaccione al entorno.

Componentes:

- Placa Arduino
- Mini servomotor
- LDR (sensor de luz)
- Resistencia ($10k\Omega$ para el LDR)
- Placa protoboard
- Cables de conexión

Pasos:

1. Conectar el LDR y el servomotor a la protoboard y al Arduino, asegurándose de utilizar la resistencia correcta, figura 5.
2. Programar el Arduino para que el servomotor ajuste su posición en función de la cantidad de luz detectada por el LDR.
3. Crear una estructura artística que se mueva junto con el servo, como una "flor robótica" que se abra o cierre en función de la luz.