



Control programado con Arduino

Práctica 8: El Relé (Control Magnético de Potencia)

Hasta el momento, todas nuestras prácticas han controlado pequeños diodos LED que consumen muy poca corriente. Sin embargo, en el mundo real, los automatismos deben activar motores, calefactores o bombillas que funcionan a voltajes elevados. Hoy aprenderemos a utilizar un **relé**, un interruptor mecánico controlado por magnetismo que nos permite gobernar grandes cargas eléctricas de forma segura y aislada.

Objetivos de Aprendizaje

Al finalizar esta sesión, serás capaz de:

- Comprender el principio de funcionamiento electromecánico de un relé.
- Diferenciar físicamente los terminales de un relé: Bobina, Común (COM), Normalmente Abierto (NO) y Normalmente Cerrado (NC).
- Diseñar un circuito con aislamiento galvánico separando la etapa de control (baja potencia) de la etapa de potencia.
- Programar secuencias temporales para la activación y desactivación de un relé magnético.

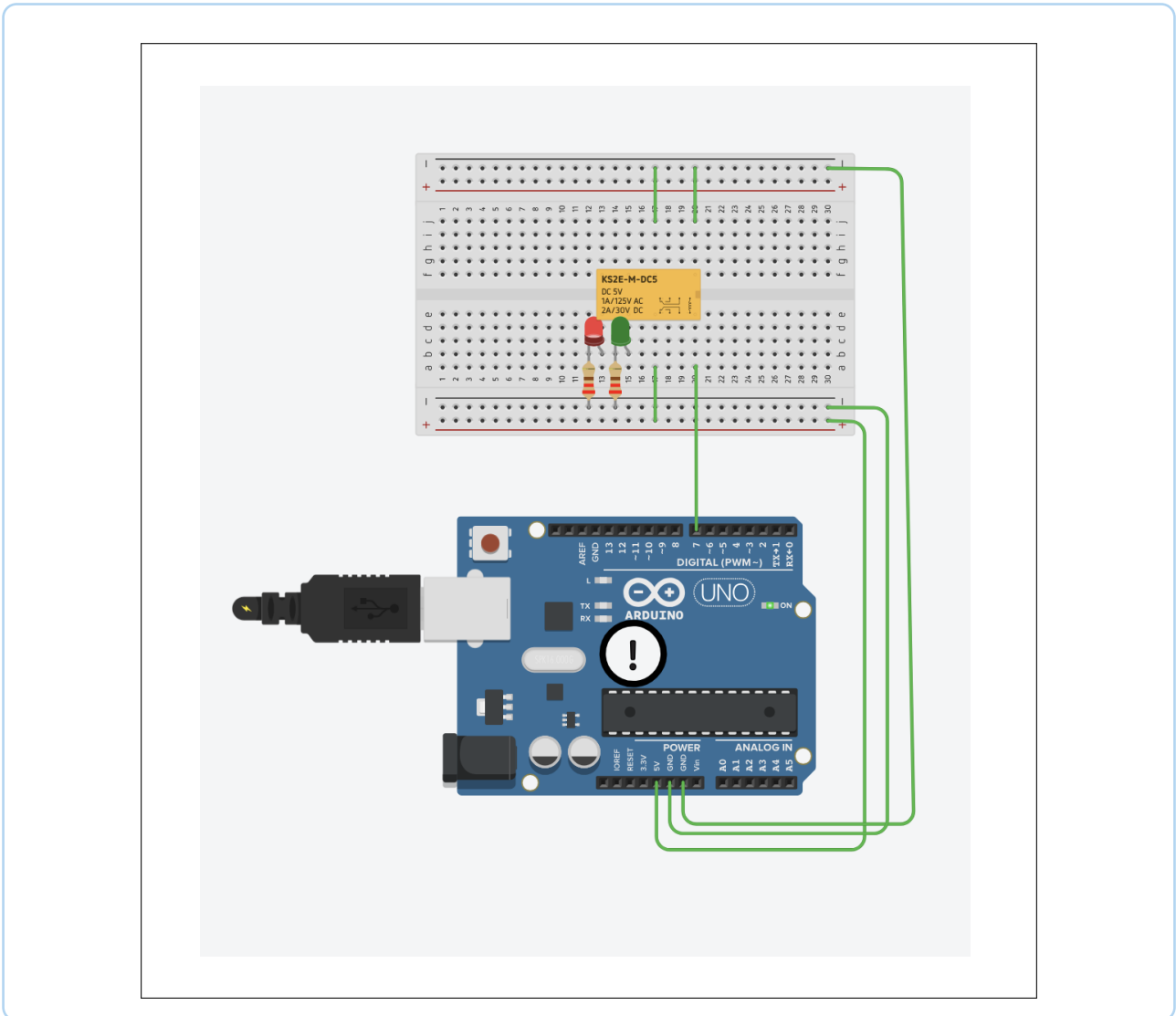
Componentes Necesarios

Busca y coloca los siguientes componentes en tu mesa de trabajo de TinkerCad:

- 1 Placa Arduino Uno.
- 1 Placa de pruebas pequeña (Protoboard).
- 1 Relé SPDT de 5 Voltios (en TinkerCad figura como "Relé SPDT").
- 1 Diodo LED (para simular la carga que vamos a encender).
- 1 Resistencia de $220\ \Omega$ (protección del LED).
- Cables de conexión virtuales.

Esquema de Montaje en la Protoboard

El relé actúa como un interruptor físico. La corriente que sale de Arduino solo se usa para activar un electroimán interno (bobina). El circuito que enciende el LED es completamente independiente.



Código Base: Activación Intermitente

Escribe el siguiente programa en el editor de TinkerCad. Verás que programar un relé es idéntico a programar un LED: solo tenemos que enviar una señal digital HIGH o LOW por el pin correspondiente.

```
1 // Asignamos el Pin Digital 7 para controlar la bobina del rele
2 int pinRele = 7;
3
4 void setup() {
5     // El pin del rele funciona como SALIDA (suministra energia a la bobina)
6     pinMode(pinRele, OUTPUT);
7 }
8
9 void loop() {
10    // ESTADO 1: Activamos el rele (atrae el contacto mecanico hacia 'NO')
11    digitalWrite(pinRele, HIGH);
12    delay(3000); // Mantenemos el rele activado durante 3 segundos
13
14    // ESTADO 2: Desactivamos el rele (el muelle interno devuelve el contacto a '
15    // NC')
16    digitalWrite(pinRele, LOW);
17    delay(3000); // Mantenemos el rele desactivado durante 3 segundos
18 }
```

Listing 1: Código para activar y desactivar un relé cíclicamente.

¿Cómo funciona el código?

Cuando inicias la simulación en TinkerCad, escucharás un chasquido visual y verás cómo la aguja interna del relé cambia de posición:

- ****Lógica del Relé****: Al ejecutar `digitalWrite HIGH`, el pin 7 envía 5V a la bobina del relé. Esto genera un campo magnético que atrae la lámina metálica del terminal ****Común (COM)****, conectándola mecánicamente con el terminal ****Normalmente Abierto (NO)****. En ese instante, la corriente fluye hacia el LED y este se enciende.
- ****El muelle de retorno****: Al hacer `digitalWrite LOW`, la bobina se queda sin electricidad, el magnetismo desaparece y un muelle interno empuja la lámina metálica de vuelta al terminal ****Normalmente Cerrado (NC)****, apagando el LED de forma instantánea.

Protección Eléctrica en la Realidad

Aunque en el simulador TinkerCad conectamos el relé directamente al pin de Arduino, en circuitos reales **nunca** se debe alimentar la bobina de un relé directamente desde un pin del microcontrolador. Las bobinas consumen más corriente de la que Arduino puede proporcionar de forma segura y generan picos de alta tensión al apagarse. En un montaje real se utiliza un transistor (como interruptor intermedio) y un diodo de protección en paralelo (diodo volante o flyback).

El Reto de la Práctica 8

Domina el uso de conmutadores electromecánicos resolviendo los siguientes retos prácticos:

1. **Indicador de Estado Doble:** Añade un segundo LED (por ejemplo, de color Rojo) al circuito. Haz las conexiones necesarias para que el LED Verde esté en el terminal **NO** y el LED Rojo en el terminal **NC**. Sin tocar el código base, observa qué sucede al iniciar la simulación. ¿Cómo se comportan los LEDs?
2. **Interruptor de Potencia Remoto:** Integra un pulsador (como el de la Práctica 3) en tu circuito de TinkerCad. Modifica el código para que el relé permanezca inactivo (apagado) y **solo se active de forma sostenida mientras mantengas presionado el pulsador**.
3. *Pregunta para reflexionar:* ¿Por qué crees que a este componente se le asocia con el concepto de "aislamiento galvánico"? ¿Qué ocurriría si la carga del circuito de potencia sufriera un cortocircuito grave? ¿Se vería afectada la placa de Arduino?