



Controlo programado com Arduino

Prática 8: O Relé (Controlo Magnético de Potência)

Até ao momento, todas as nossas práticas controlaram pequenos díodos LED que consomem muito pouca corrente. No entanto, no mundo real, os automatismos devem ativar motores, aquecedores ou lâmpadas que funcionam a tensões elevadas. Hoje aprenderemos a utilizar um **relé**, um interruptor mecânico controlado por magnetismo que nos permite governar grandes cargas elétricas de forma segura e isolada.

Objetivos de Aprendizagem

Ao finalizar esta sessão, serás capaz de:

- Compreender o princípio de funcionamento eletromecânico de um relé.
- Diferenciar fisicamente os terminais de um relé: Bobina, Comum (COM), Normalmente Aberto (NO) e Normalmente Fechado (NC).
- Desenhar um circuito com isolamento galvânico separando a etapa de controlo (baixa potência) da etapa de potência.
- Programar sequências temporais para a ativação e desativação de um relé magnético.

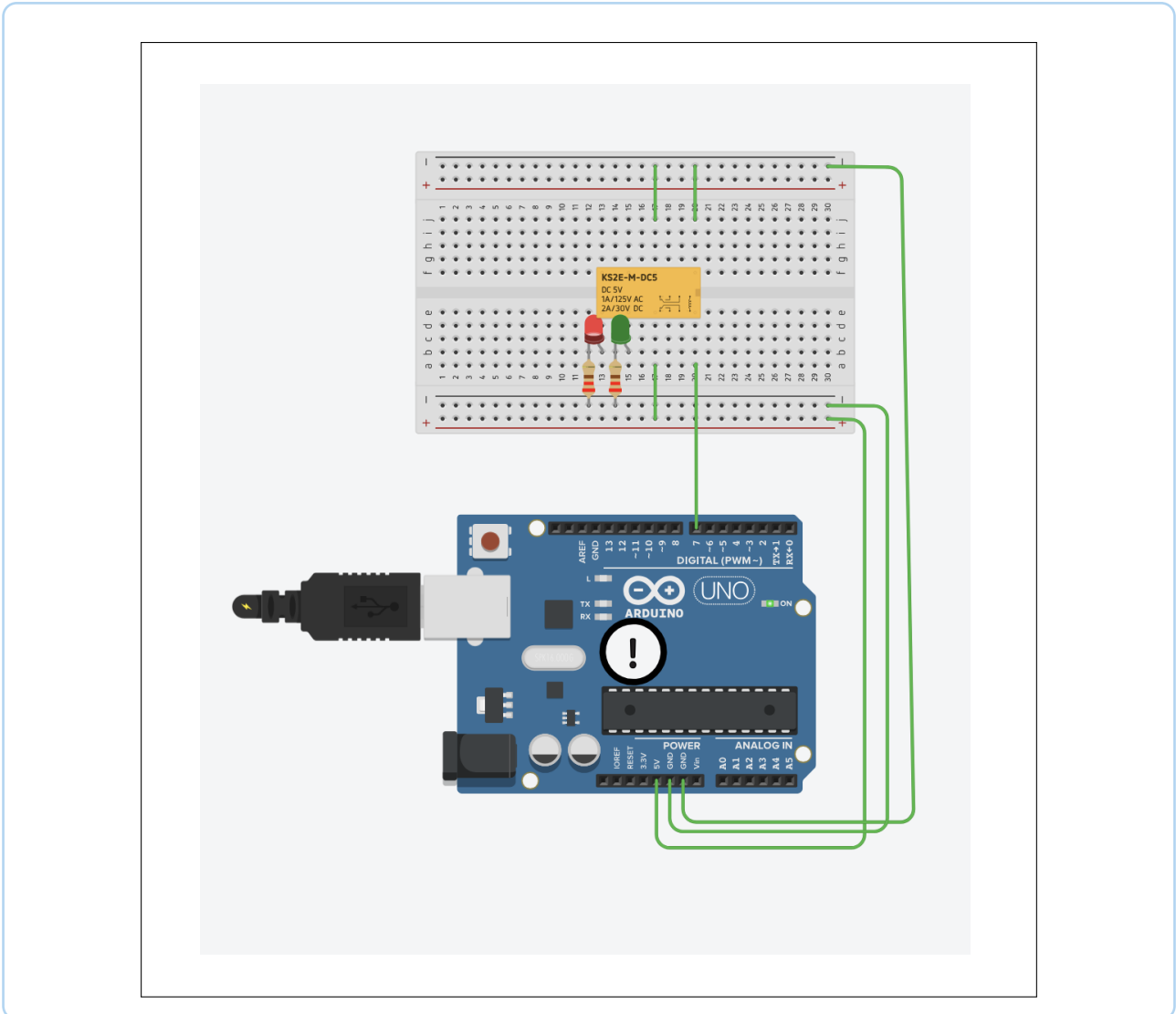
Componentes Necessários

Procura e coloca os seguintes componentes na tua mesa de trabalho do TinkerCad:

- 1 Placa Arduino Uno.
- 1 Placa de ensaio pequena (Protoboard).
- 1 Relé SPDT de 5 Volts (no TinkerCad aparece como "Relé SPDT").
- 1 Díodo LED (para simular a carga que vamos acender).
- 1 Resistência de $220\ \Omega$ (proteção do LED).
- Cabos de ligação virtuais.

Esquema de Montagem na Protoboard

O relé atua como um interruptor físico. A corrente que sai do Arduino apenas é usada para ativar um eletroímã interno (bobina). O circuito que acende o LED é completamente independente.



Código Base: Ativação Intermitente

Escreve o seguinte programa no editor do TinkerCad. Verás que programar um relé é idêntico a programar um LED: apenas temos de enviar um sinal digital HIGH ou LOW pelo pino correspondente.

```
1 // Atribuimos o Pino Digital 7 para controlar a bobina do rele
2 int pinRele = 7;
3
4 void setup() {
5     // O pino do rele funciona como SAIDA (fornece energia a bobina)
6     pinMode(pinRele, OUTPUT);
7 }
8
9 void loop() {
10    // ESTADO 1: Ativamos o rele (atrai o contacto mecanico para 'NO')
11    digitalWrite(pinRele, HIGH);
12    delay(3000); // Mantemos o rele ativado durante 3 segundos
13
14    // ESTADO 2: Desativamos o rele (a mola interna devolve o contacto a 'NC')
15    digitalWrite(pinRele, LOW);
16    delay(3000); // Mantemos o rele desativado durante 3 segundos
17 }
```

Listing 1: Código para ativar e desativar um relé ciclicamente.

Como funciona o código?

Quando inicias a simulação no TinkerCad, ouvirás um clique visual e verás como a agulha interna do relé muda de posição:

- ****Lógica do Relé****: Ao executar `digitalWrite` `HIGH`, o pino 7 envia 5V para a bobina do relé. Isto gera um campo magnético que atrai a lâmina metálica del terminal ****Comum (COM)****, ligando-a mecanicamente ao terminal ****Normalmente Aberto (NO)****. Nesse instante, a corrente flui para o LED e este acende-se.
- ****A mola de retorno****: Ao fazer `digitalWrite` `LOW`, a bobina fica sem eletricidade, o magnetismo desaparece e uma mola interna empurra a lâmina metálica de volta ao terminal ****Normalmente Fechado (NC)****, apagando o LED de forma instantânea.

Proteção Elétrica na Realidade

Embora no simulador TinkerCad liguemos o relé diretamente ao pino do Arduino, em circuitos reais **nunca** se deve alimentar a bobina de um relé diretamente a partir de um pino do microcontrolador. As bobinas consomem mais corrente do que o Arduino consegue fornecer em segurança e geram picos de alta tensão ao desligar. Numa montagem real, utiliza-se um transístor (como interruptor intermédio) e um díodo de proteção em paralelo (díodo volante ou flyback).

O Desafio da Prática 8

Domina o uso de comutadores eletromecânicos resolvendo os seguintes desafios práticos:

1. **Indicador de Estado Duplo:** Adiciona um segundo LED (por exemplo, de cor Vermelha) ao circuito. Faz as ligações necessárias para que o LED Verde esteja no terminal **NO** e o LED Vermelho no terminal **NC**. Sem alterar o código base, observa o que acontece ao iniciar a simulação. Como se comportam os LEDs?
2. **Interruptor de Potência Remoto:** Integra um botão de pressão (como o da Prática 3) no teu circuito do TinkerCad. Modifica o código para que o relé permaneça inativo (desligado) e **apenas se ative de forma sustentada enquanto mantiveres o botão pressionado**.
3. *Pergunta para refletir:* Por que razão achas que este componente é associado ao conceito de “isolamento galvânico”? O que aconteceria se a carga do circuito de potência sofresse um curto-circuito grave? A placa do Arduino seria afetada?