



Controlo programado com Arduino

Prática 6: Luz sequencial (Estruturas Iterativas)

Nas práticas anteriores aprendemos a sequenciar um semáforo escrevendo instruções linha a linha. No entanto, em engenharia e programação profissional, repetir blocos idênticos de código é ineficiente. Nesta sessão ligaremos 5 LEDs e aprenderemos a utilizar o ciclo iterativo `for` para os governar de forma ágil, reduzindo drasticamente o tamanho do nosso programa.

Objetivos de Aprendizagem

Ao finalizar esta sessão, serás capaz de:

- Cablar um barramento linear de múltiplas saídas digitais (5 LEDs em paralelo) na protoboard.
- Compreender a lógica e a estrutura interna da instrução iterativa `for`.
- Utilizar variáveis locais de controlo (como a variável índice) para direcionar pinos físicos.
- Otimizar o código reduzindo a redundância de instruções repetitivas.

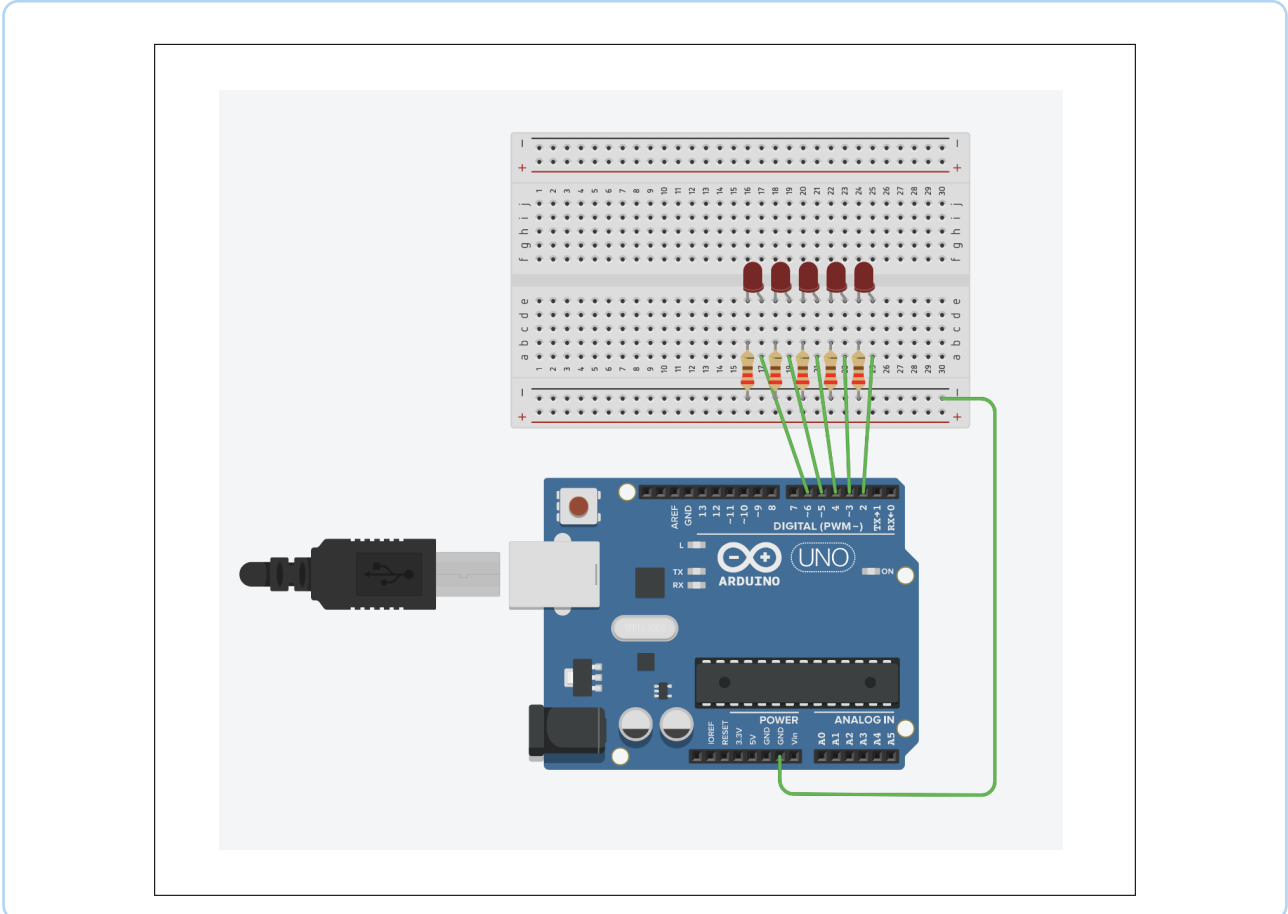
Componentes Necessários

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

- 1 Placa Arduino Uno.
- 1 Placa de ensaio pequena (Protoboard).
- 5 Díodos LED (sugerimos que uses cores variadas).
- 5 Resistências de $220\ \Omega$ (indispensável colocar uma para cada LED).
- Cabos de ligação virtuais.

Esquema de Montagem na Protoboard

Ligaremos os 5 LEDs a 5 pinos digitais consecutivos do Arduino (do pino 2 ao pino 6). Isto facilitará muito o desenho lógico do ciclo numérico.



Código Base: Inicialização e Varrimento Lineal

Abre o editor de código no TinkerCad. Escreve o seguinte programa para observar como um ciclo `for` se encarrega de configurar todos os pinos e outro realiza um varrimento de acendimento contínuo.

```
1 // Pratica 6: Varrimento sequencial de 5 LEDs otimizado com ciclos for
2
3 void setup() {
4   // Configura os pinos do 2 ao 6 como saidas usando um unico ciclo
5   // A variavel 'i' tomara os valores: 2, 3, 4, 5 e 6
6   for (int i = 2; i <= 6; i++) {
7     pinMode(i, OUTPUT);
8   }
9 }
10
11 void loop() {
12   // Varrimento da esquerda para a direita
13   for (int i = 2; i <= 6; i++) {
14     digitalWrite(i, HIGH); // Acende o LED ligado ao pino atual 'i'
15     delay(150);           // Pausa breve de 150 milissegundos
16     digitalWrite(i, LOW); // Apaga o LED atual antes de passar ao seguinte
17   }
18 }
```

Listing 1: Código para realizar um varrimento linear sobre 5 LEDs.

Como funciona o código?

O ciclo `for` é uma estrutura de controlo que repete um bloco de código um número exato de vezes. O seu cabeçalho consiste em três partes chave separadas por pontos e vírgulas:

- **Inicialização** (`int`): Cria uma variável temporária chamada contador () e atribui-lhe o valor do primeiro pino físico que usaremos (2).
- **Condição de paragem** (): Antes de cada volta, o Arduino verifica se esta condição matemática é verdadeira. Se `for` for menor ou igual a 6, executa o código entre as chavetas. Assim que `for` passa a valer 7, o ciclo termina.
- **Atualização ou incremento** (): Ao terminar cada iteração, somamos 1 ao valor atual da variável (`for` equivale a escrever `i = i + 1`).

Durante cada passo do ciclo, a ordem `digitalWrite HIGH` substitui a variável `for` pelo valor numérico correspondente nesse instante, alcançando o movimento físico da luz.

O Desafio da Prática 6

Sobe mais um degrau na escala de aprendizagem resolvendo os seguintes desafios lógicos no teu simulador:

1. **Varrimento de duplo sentido (O Carro Fantástico):** Adiciona um segundo ciclo `for` dentro da função `loop` para que a sequência faça um percurso de volta (da direita para a esquerda). *Pista: Debes começar o contador em 5, verificar que seja maior ou igual a 3 utilizando `>=`, e ir subtraindo uma unidade em cada passo com `-`.*
2. **Efeito cumulativo:** Modifica o código para que os LEDs se vão acendendo um a um sequencialmente mas **não se apaguem de imediato**, mas sim que fiquem todos acesos e depois se apaguem todos ao mesmo tempo ao finalizar a ronda.
3. *Pergunta para refletir:* Se no futuro decidirmos ampliar a nossa montagem adicionando mais três LEDs nos pinos 7, 8 e 9, que valores concretos do cabeçalho dos nossos ciclos `for` teríamos de modificar no código?