



Controlo programado com Arduino

Prática 3: O Botão de Pressão Pulsador (Funcionamento Monestável)

Até agora, os nossos circuitos apenas enviavam informação para o exterior acendendo LEDs (canais de saída). Nesta prática daremos um passo crucial: aprenderemos a receber dados do exterior utilizando um botão de pressão (canal de entrada). Desenharemos um programa inteligente que tome uma decisão lógica e acenda um LED apenas enquanto mantivermos o botão pressionado.

Objetivos de Aprendizagem

Ao finalizar esta sessão, serás capaz de:

- Configurar pinos digitais como canais de entrada de dados utilizando `pinMode` `INPUT` .
- Ler e monitorizar estados digitais (valores lógicos HIGH e LOW) com o comando `digitalRead` .
- Aplicar estruturas de controlo condicionais através da instrução `if else`.
- Entender a necessidade elétrica de utilizar uma resistência de proteção (Pull-Down) para evitar o estado flutuante (ruído elétrico).

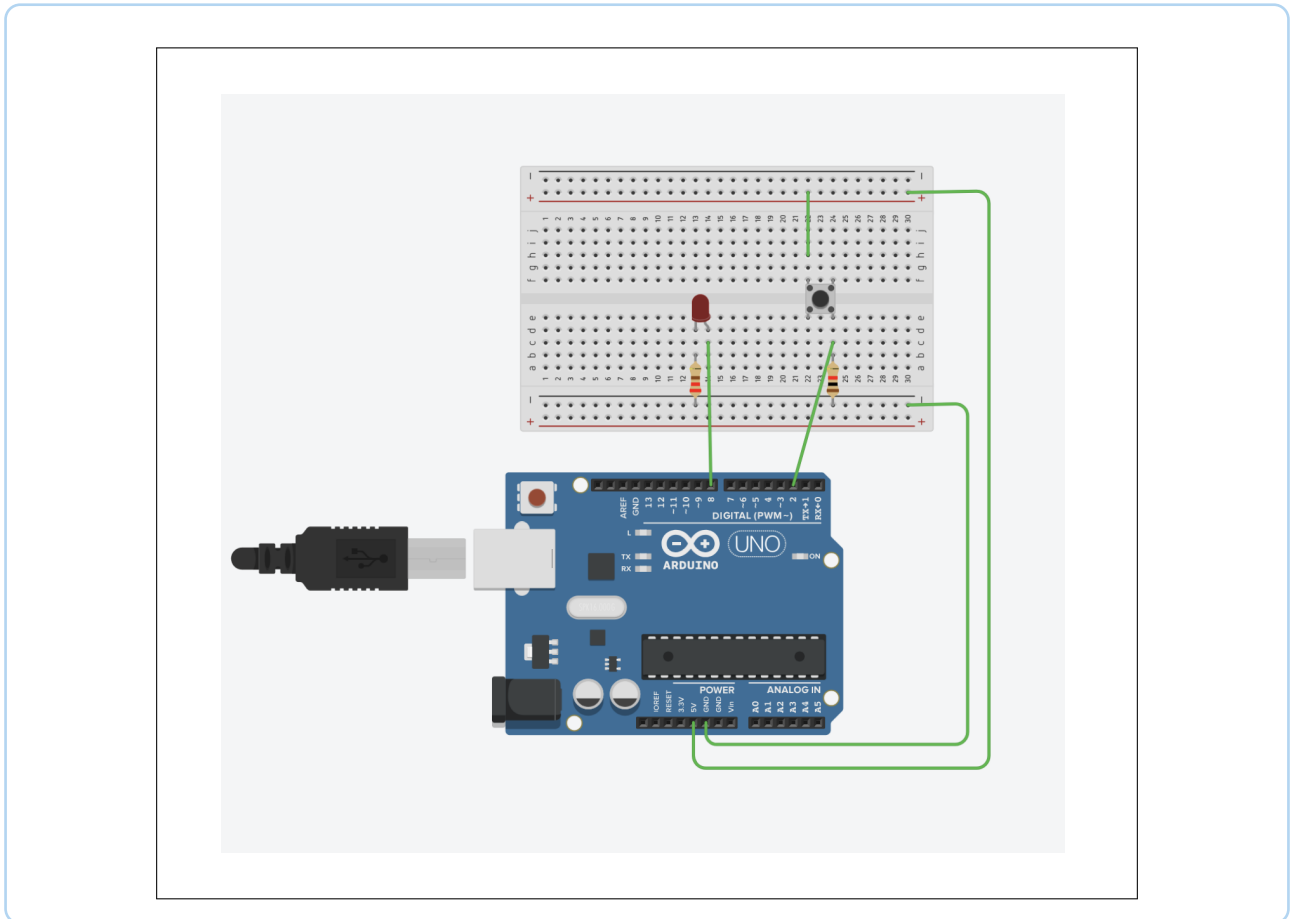
Componentes Necessários

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

- 1 Placa Arduino Uno.
- 1 Placa de ensaio pequena (Protoboard).
- 1 Díodo LED e 1 resistência de 220 Ω (para o LED).
- 1 Botão de pressão de quatro pinos.
- 1 Resistência de 10 k Ω (código de cores: castanho, preto, laranja; será usada como Pull-Down).
- Cabos de ligação virtuais.

Esquema de Montagem na Protoboard

É fundamental ligar o botão de pressão de forma adequada. Para que o Arduino leia um zero elétrico (0V) quando ninguém pressionar o botão, utilizaremos uma configuração de resistência em **Pull-Down**.



Código Base: O condicional if/else

Escreve o seguinte programa no editor de código do TinkerCad. Observa com atenção como o código decide de maneira contínua o estado do LED lendo a tensão do botão de pressão.

```
1 // Declaracao de variaveis atribuindo os pinos de ligacao
2 int led = 8;           // O LED esta ligado no pino digital 8
3 int pulsador = 2;     // O botao de pressao esta ligado no pino digital 2
4
5 void setup() {
6   pinMode(led, OUTPUT);           // Configuramos o LED como SAIDA de corrente
7   pinMode(pulsador, INPUT);      // Configuramos o botao de pressao como ENTRADA de
   dados
8 }
9
10 void loop() {
11   // Criamos uma variavel local para guardar o estado do botao (HIGH ou LOW)
12   int estadoPulsador = digitalRead(pulsador);
13
14   // Condicional: Se o botao estiver pressionado (recebe 5V / HIGH)
15   if (estadoPulsador == HIGH) {
16     digitalWrite(led, HIGH);    // Acende o LED
17   }
18   // Caso contrario (se nao estiver pressionado / recebe 0V / LOW)
19   else {
20     digitalWrite(led, LOW);     // Apaga o LED
21   }
22 }
```

Listing 1: Código para controlar um LED com um botão de pressão momentâneo.

Como funciona o código?

- `pinMode INPUT` : Indica ao microcontrolador que o pino 2 não vai fornecer corrente, mas sim "escutar" de forma passiva a tensão que lhe chegue do circuito exterior.
- `digitalRead` : É um sensor lógico. Lê a tensão no pino indicado. Se detetar 5V devolverá o valor `HIGH` (alto), e se detetar 0V (GND) devolverá `LOW` (baixo).
- **A comparação** : Em programação, um único símbolo de igual (`=`) serve para guardar um valor dentro de uma variável. Para comparar se duas coisas são matematicamente iguais, devemos usar obrigatoriamente o duplo igual (`==`).
- **if else (Se / Caso contrário)**: Avalia uma condição lógica entre parênteses. Se a condição se cumprir, executam-se as linhas delimitadas pelas chavetas do `if`. Se a condição for falsa, o programa ignora esse bloco e executa diretamente o que está dentro das chavetas do `else`.

O Desafio da Prática 3

Demonstra a tua compreensão sobre o controlo do fluxo do código superando estes três desafios:

1. **Lógica Inversa:** Modifica o programa no TinkerCad para que o LED esteja sempre aceso por defeito, e se **apague** apenas quando mantiveres o botão de pressão pressionado.
2. **Semáforo de Peões de duas posições:** Adiciona um segundo LED (por exemplo, verde no pino 9 e vermelho no pino 8). Faz com que, por defeito, o LED vermelho esteja aceso e o verde apagado. Ao pressionar o botão, o vermelho deve apagar-se e o verde acender-se.
3. *Pergunta para refletir:* Se retirarmos a resistência de $10\text{ k}\Omega$ da protoboard e ligarmos o botão diretamente ao GND e aos 5V sem ela, que tipo de anomalia elétrica ou física achas que poderia acontecer à placa do Arduino ao pressionar o botão?