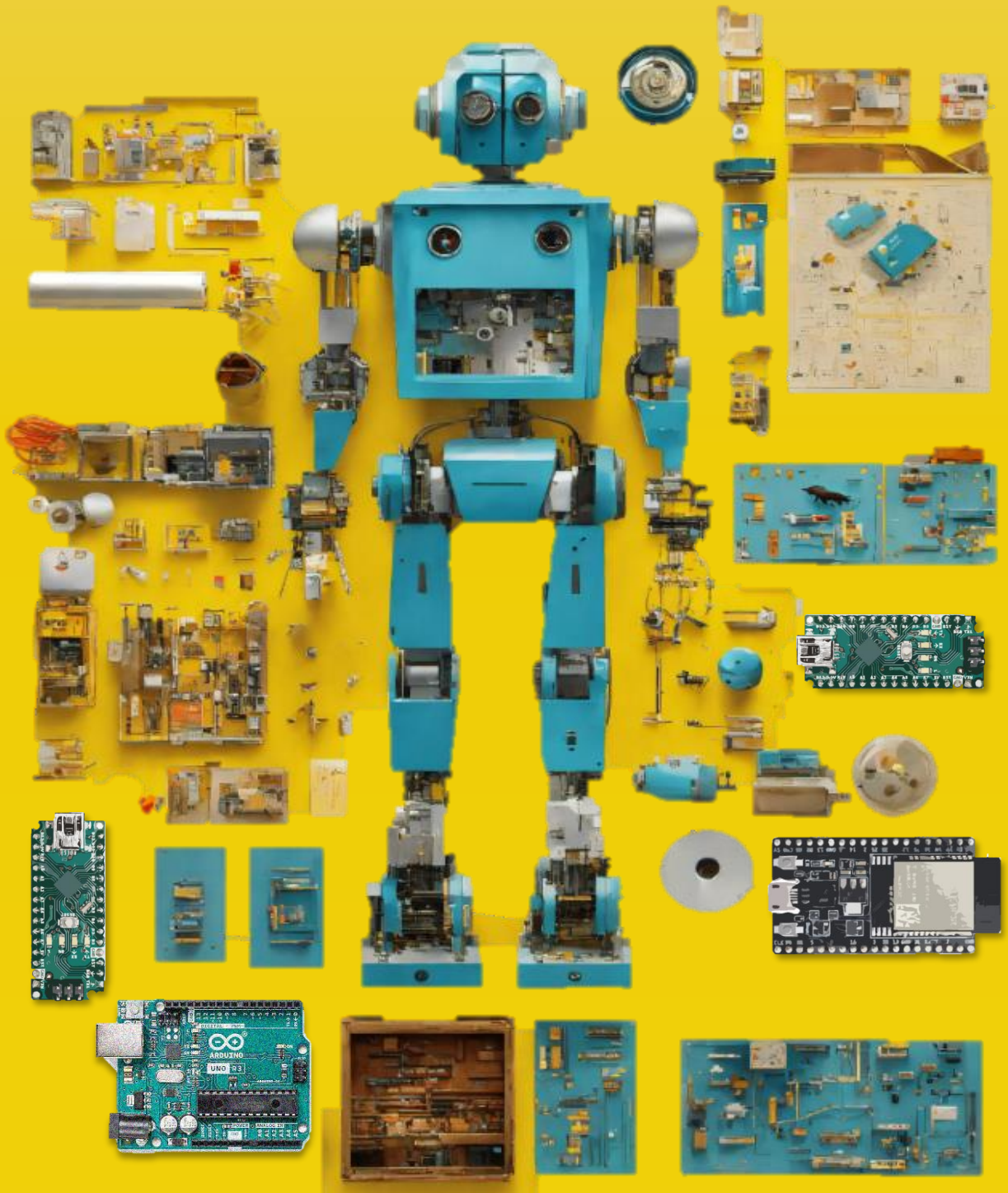


Arduino

o guia do mochileiro
da Robótica!



Por M.M.Takuno

01

Introdução

Introdução

O **Arduino** é uma plataforma de prototipagem eletrônica de código aberto, ideal para quem deseja aprender sobre eletrônica e programação. Com ele, você pode criar projetos incríveis, desde um simples piscar de LED até complexos sistemas automatizados.

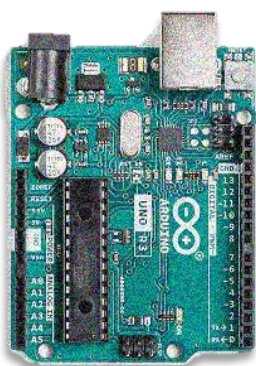
Neste ebook, vamos explorar os conceitos básicos do Arduino e apresentar exemplos de código para ajudá-lo a começar sua jornada.



Sobre o Arduino

O Arduino é uma placa com um microcontrolador que pode ser programado para controlar componentes eletrônicos.

A placa mais comum é a Arduino Uno, mas existem várias outras, como a Mega e a Nano.



Arduino
Uno R3



Arduino
micro



Arduino
nano

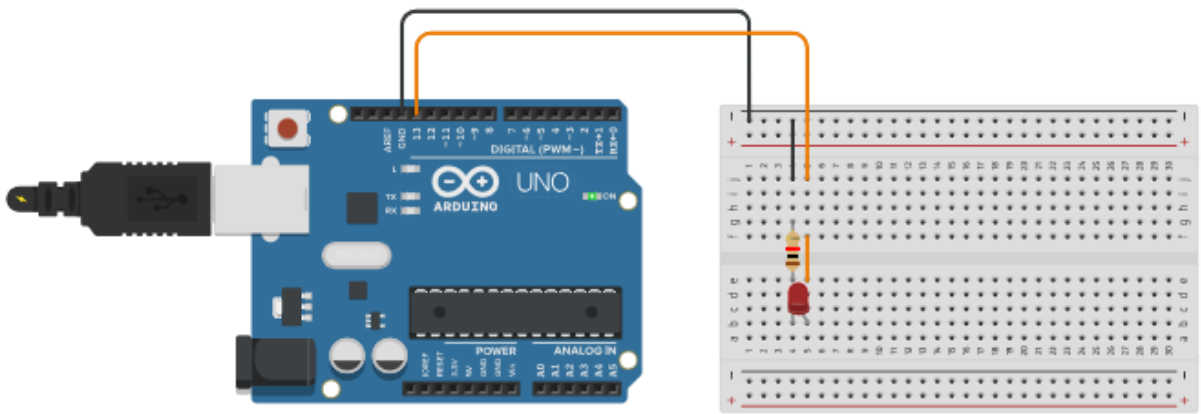
O Arduino é programado usando a linguagem C/C++ no ambiente de desenvolvimento Arduino IDE.

02

Componentes

Piscar um LED

Um dos primeiros projetos com Arduino é fazer um LED piscar. É simples, mas ensina os conceitos fundamentais de controle de saídas digitais.



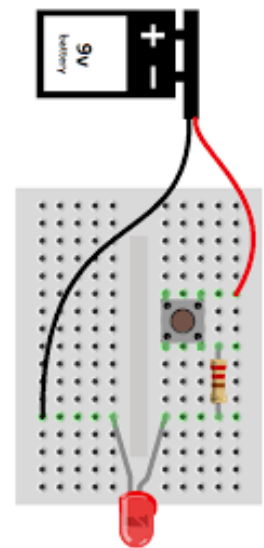
Exemplo de código:

```
void setup() {  
  pinMode(13, OUTPUT); // Configura o pino 13 (digital) como saída  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // Liga o LED  
  delay(1000); // Espera 1 segundo  
  digitalWrite(13, LOW); // Desliga o LED  
  delay(1000); // Espera 1 segundo  
}
```

Lendo um Botão

Além de controlar saídas, o Arduino pode ler entradas, sejam elas analógicas ou digitais. Vamos usar um botão (sinal digital) para acionar um LED.

O circuito conecta um botão ao pino 2 e ao GND do Arduino. Quando pressionado, o botão envia um sinal, que o Arduino usa para acionar um LED.



Exemplo de código:

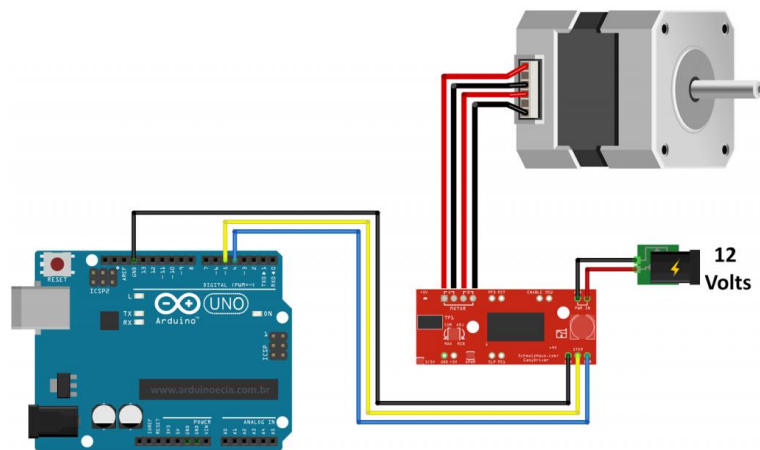
```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop() {
  buttonState = digitalRead(buttonPin); // Lê o estado do botão
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH); // Liga o LED com botão pressionado
  } else {
    digitalWrite(ledPin, LOW); // Desliga o LED com botão não-pressionado
  }
}
```

Controlando Motores

Para controlar a velocidade de um motor, usamos a saída PWM (Pulse Width Modulation) do Arduino, uma saída analógica (diferente da saída digital que utilizamos anteriormente no botão).



Exemplo de código:

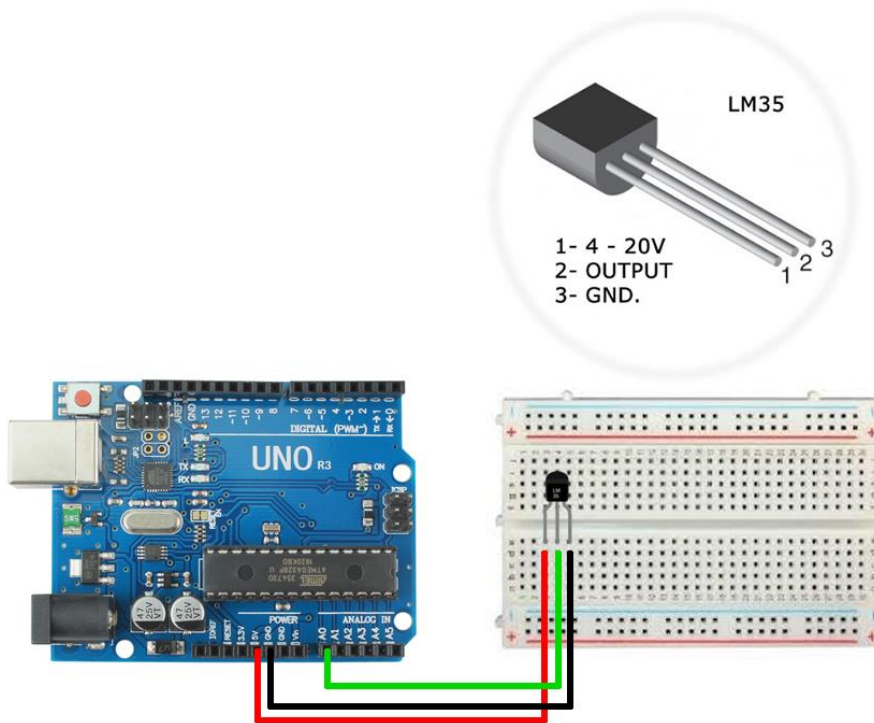
```
const int motorPin = 9; // Pino PWM

void setup() {
  pinMode(motorPin, OUTPUT);
}

void loop() {
  analogWrite(motorPin, 128); // Define a velocidade do motor (0-255)
  delay(2000); // Espera 2 segundos
  analogWrite(motorPin, 0); // Desliga o motor
  delay(2000); // Espera 2 segundos
}
```


Sensores: Lendo dados

Usando um sensor de temperatura LM35, podemos ler a temperatura ambiente e exibi-la no monitor serial.



Exemplo de código:

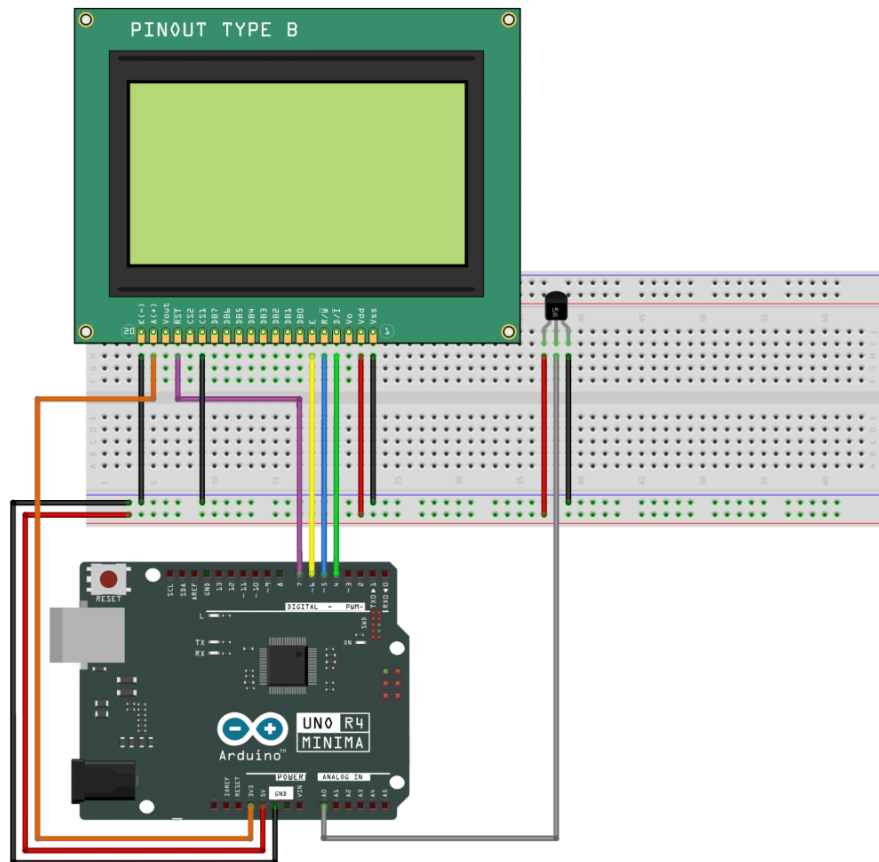
```
const int sensorPin = A0;

void setup() {
  Serial.begin(9600); // Inicia a comunicação serial
}

void loop() {
  int sensorValue = analogRead(sensorPin); // Lê o valor do sensor
  float temperature = (sensorValue * 5.0 * 100.0) / 1024.0; // Converte para temperatura
  Serial.println(temperature); // Exibe a temperatura no monitor serial
  delay(1000); // Espera 1 segundo
}
```

Display LCD

Para exibir informações, podemos usar um display LCD. Vamos mostrar a temperatura lida pelo sensor LM35.:



fritzing

Display LCD

Exemplo de código:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
const int sensorPin = A0;

void setup() {
  lcd.begin(16, 2); // Configura o LCD com 16 colunas e 2 linhas
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(sensorPin);
  float temperature = (sensorValue * 5.0 * 100.0) / 1024.0;
  lcd.clear(); // Limpa a tela do LCD
  lcd.setCursor(0, 0); // Define o cursor na primeira linha
  lcd.print("Temp: ");
  lcd.print(temperature); // Exibe a temperatura
  lcd.print(" C");
  delay(1000);
}
```

03

Conclusão

Nova Jornada

Esses são apenas alguns conceitos básicos do Arduino. Com eles, você pode começar a explorar e criar projetos mais complexos. O importante é continuar experimentando e aprendendo.

Boa sorte na sua jornada com o Arduino!

Espero que esses exemplos práticos e explicações simples ajudem você a dar os primeiros passos no mundo do Arduino. Obrigada por ter lido até aqui e até a próxima!

