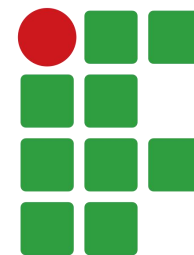


Eletricidade e Eletrônica

Curso Técnico em Informática para Internet

Lucas Sampaio Leite

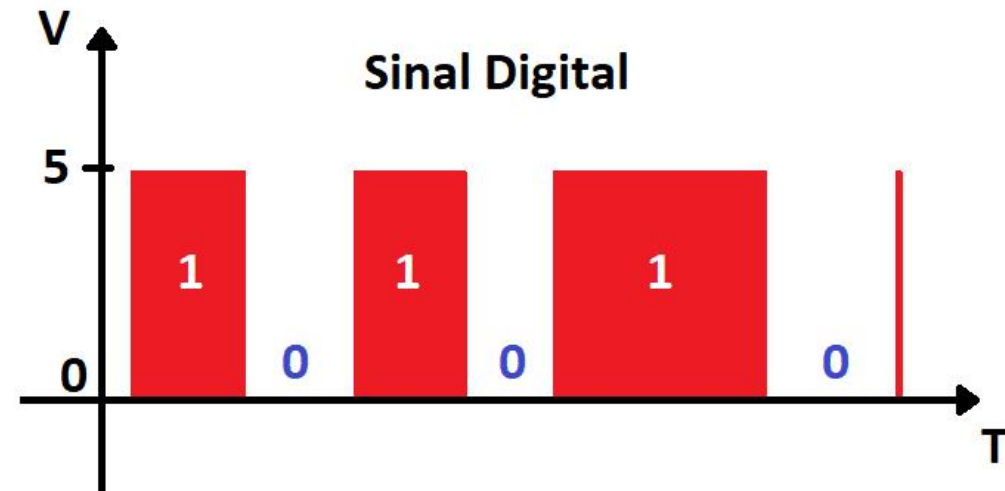
Baseada na aula do professor Elton B. Torres



**INSTITUTO
FEDERAL**
Pernambuco

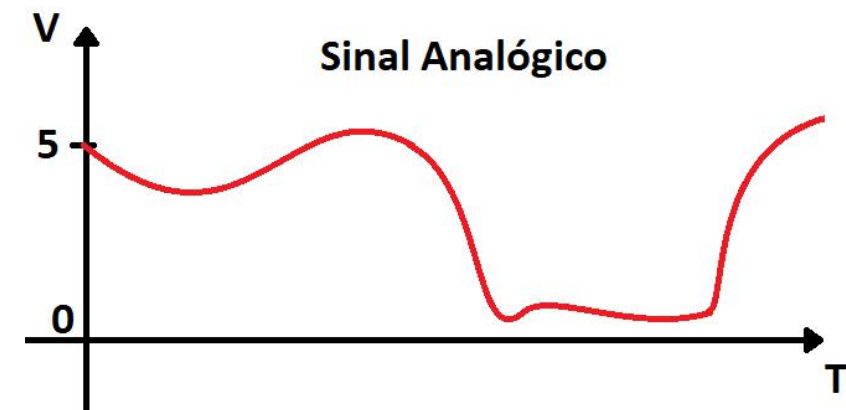
Entradas e saídas digitais

- As entradas e saídas digitais permitem representar apenas dois estados lógicos: ativado (HIGH ou 1) e desativado (LOW ou 0).
- Exemplos de sinais digitais:
 - Botões;
 - Sensores de presença;
 - Chaves liga/desliga;
 - Estados lógicos (0 e 1).



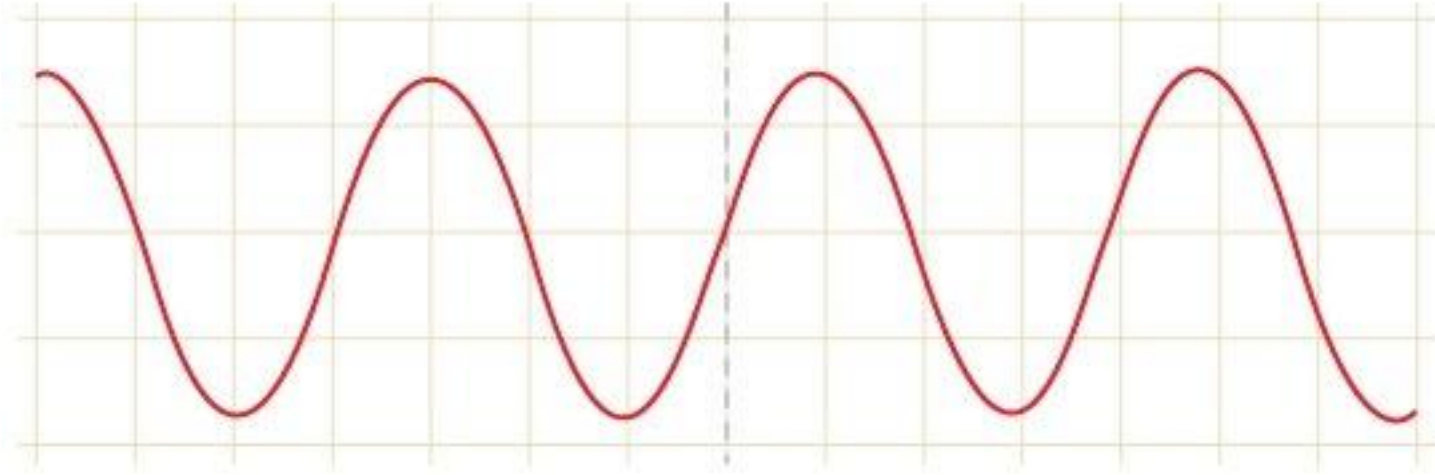
Entradas analógicas

- As entradas e saídas analógicas permitem representar uma faixa contínua de valores entre dois extremos, possibilitando medir ou controlar grandezas que variam gradualmente.
- Exemplos de sinais analógicos:
 - Sensores de temperatura (ex.: 0 a 100 °C);
 - Sensores de umidade (0 a 100%);
 - Sensores de pressão;
 - Sensores de luminosidade;
 - Potenciômetros.

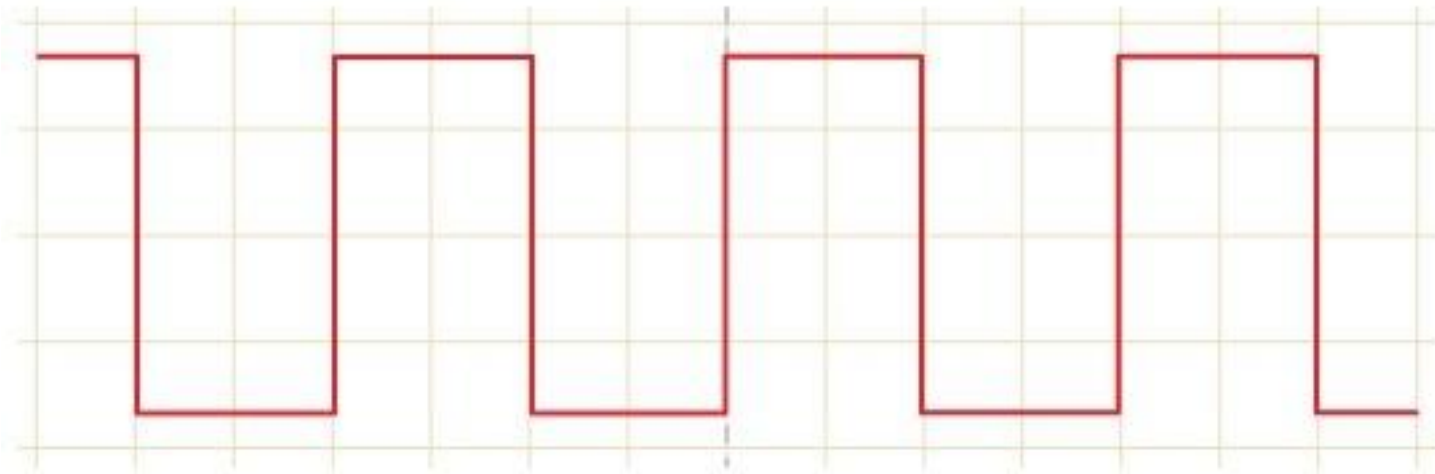


Sinal analógico vs sinal digital

- Sinal analógico:

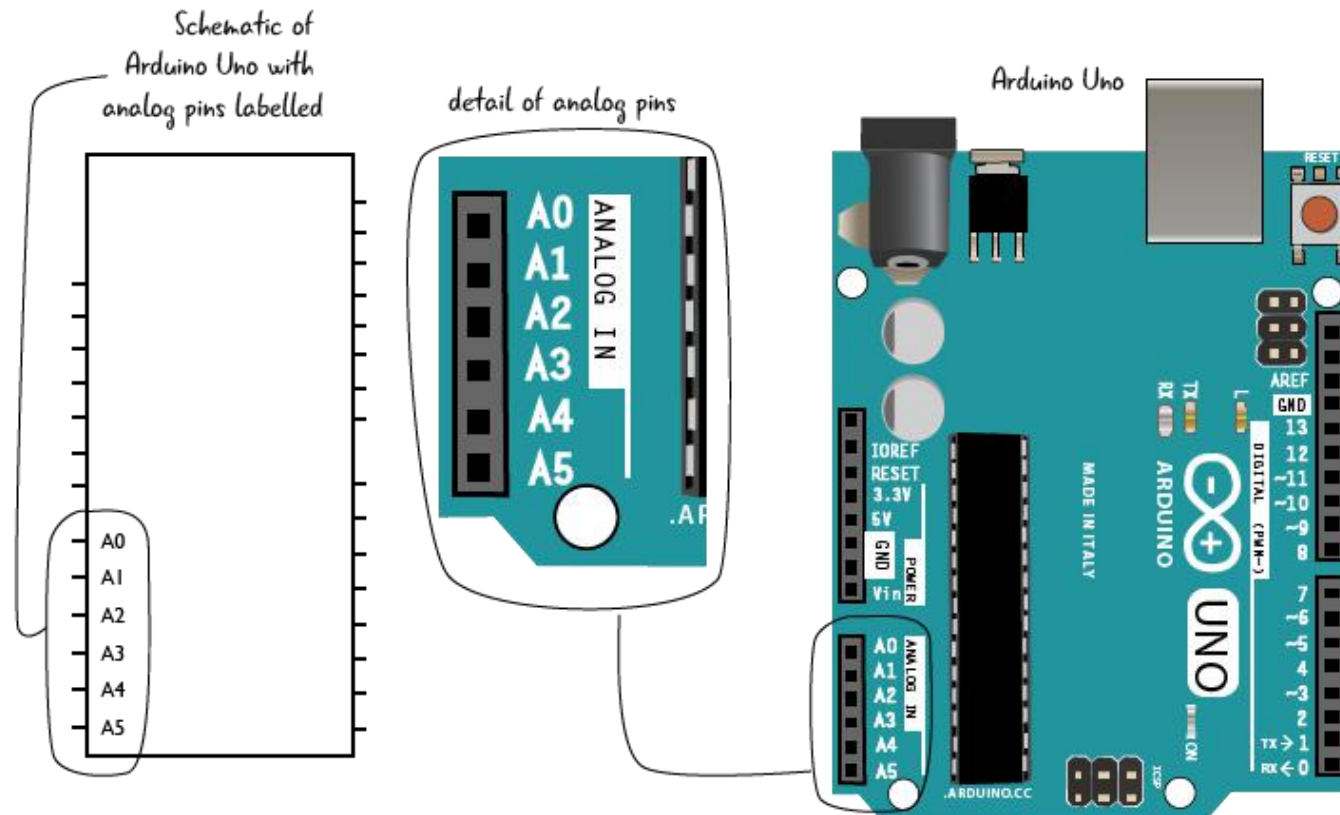


- Sinal digital:



Entradas analógicas

- Pinos de entrada análogica no Arduino Uno:



Entradas analógicas

- Lendo um valor analógico:

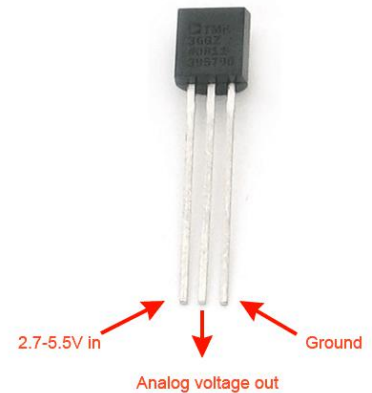
```
void loop() {  
    int valor = analogRead(entradaAnalogica); //0 a 1023  
    delay(1000);  
}
```

Entradas analógicas

- Os pinos de entrada analógica permitem ler valores entre 0 e 1023, devido ao conversor analógico-digital (A/D) de 10 bits do Arduino.
- Esse intervalo de valores possibilita medir variações contínuas de grandezas físicas, permitindo criar aplicações mais sofisticadas do que simplesmente ligar ou desligar um dispositivo.
- Entradas analógicas são utilizadas para ler a tensão elétrica fornecida por componentes, geralmente sensores, que podem gerar valores contínuos entre 0 V e 5 V.

Entradas analógicas

- O TMP36 é um sensor de temperatura analógico bastante utilizado em projetos com Arduino. Ele mede a temperatura ambiente e fornece uma tensão proporcional à temperatura, que pode ser lida por uma entrada analógica do Arduino.
- Características principais
 - Mede temperatura ambiente.
 - Fornece saída analógica (não digital).
 - Funciona com alimentação de 2,7 V a 5,5 V.
 - No Tinkercad, a temperatura pode ser alterada manualmente durante a simulação.



Entradas analógicas

- O TMP36 gera uma tensão proporcional à temperatura:

- $0\text{ }^{\circ}\text{C} \rightarrow 0,5\text{ V}$
- $25\text{ }^{\circ}\text{C} \rightarrow 0,75\text{ V}$
- $100\text{ }^{\circ}\text{C} \rightarrow 1,5\text{ V}$

- A relação é dada por:

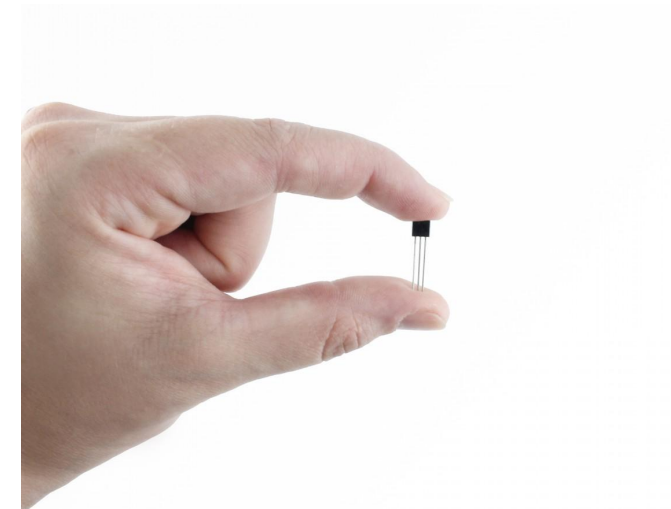
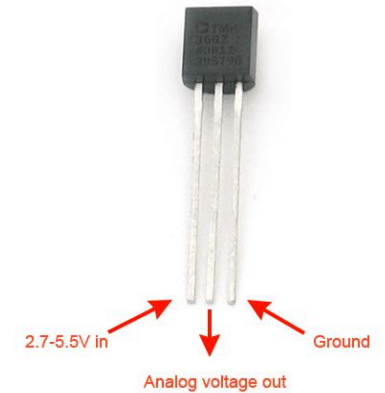
$$\text{Temperatura}(^{\circ}\text{C}) = (\text{Tensão} - 0,5) \times 100$$

- Para converter a leitura em tensão:

- $\text{float tensao} = \text{leitura} * (5.0 / 1023.0);$

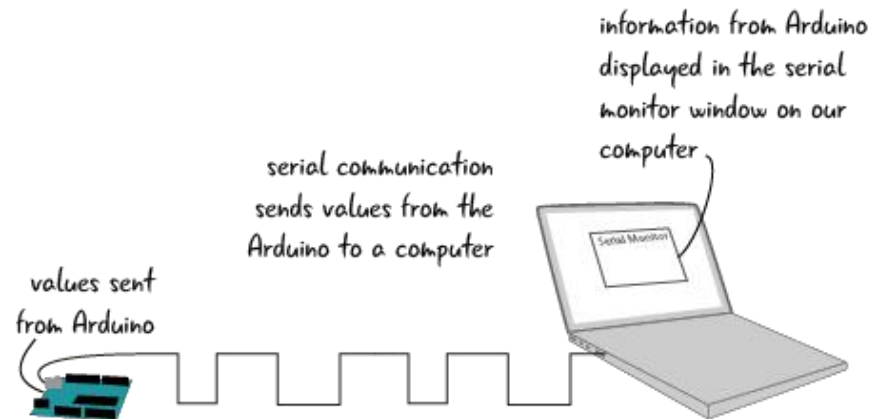
- Para converter a tensão em temperatura:

- $\text{float temperatura} = (\text{tensao} - 0.5) * 100;$



Comunicação Serial

- Comunicação Serial: É um modo de comunicação entre dispositivos. Alterando os valores de tensão no fio (HIGH e LOW) o Arduino pode transmitir informações pelo cabo USB para o computador;
- Utilizando os pinos digitais 0 e 1 o Arduino pode enviar e receber dados de/para outros dispositivos. TX envia e RX recebe.



Comunicação Serial

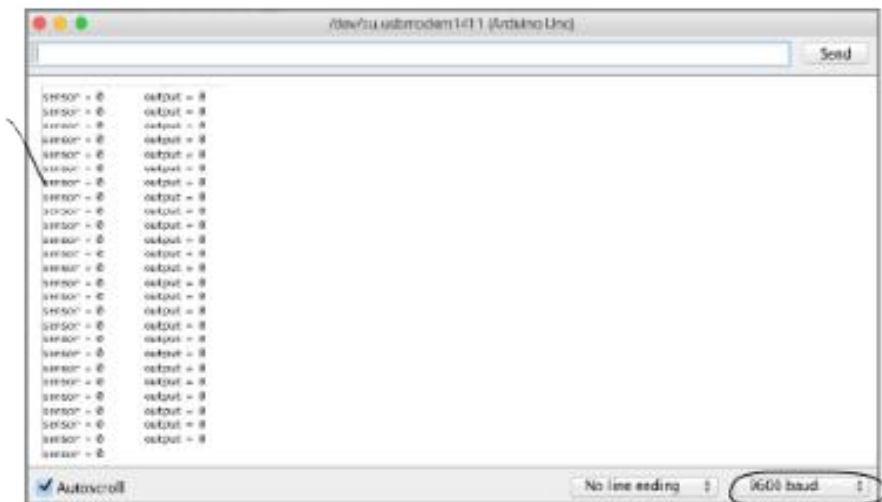
- O monitor serial é um recurso da IDE do Arduino que exibe as informações enviadas pelo Arduino. Útil para depuração e para saber quais os valores um sensor ou variável geram.



- A taxa de bits usada para comunicação entre o Arduino e o computador é 9600bps.

our setup code setting the baud rate

```
void setup() {  
  // initialize serial communications at 9600 bps:  
  Serial.begin(9600);  
}
```



Comunicação Serial

- O objeto serial tem duas funções para enviar informações para o computador: `Serial.print()` e `Serial.println()`.

code from loop for printing to our computer

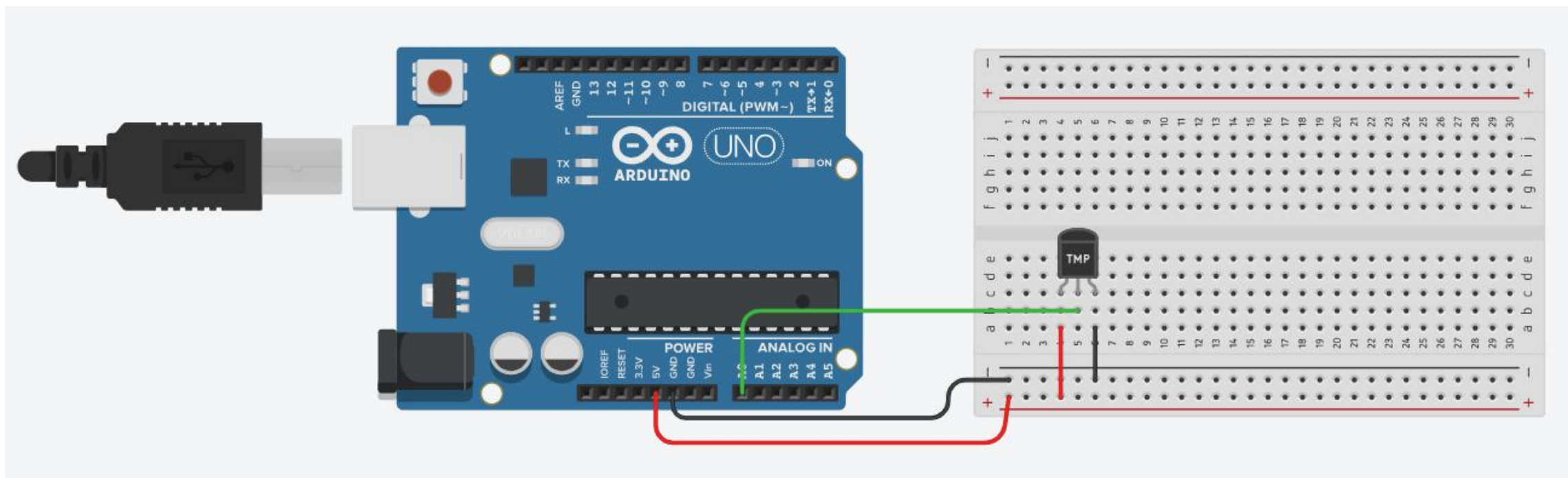
```
Serial.print("sensor = ");  
Serial.print(sensorValue);  
Serial.print("\t output = ");  
Serial.println(outputValue);
```

- Para enviar palavras ao monitor serial, utilizamos string, que é uma sequência de caracteres usadas para representar uma informação textual.

*This is how all four lines
of code will appear in
the serial monitor.*

sensor = 302	output = 75
sensor = 303	output = 75
sensor = 306	output = 76

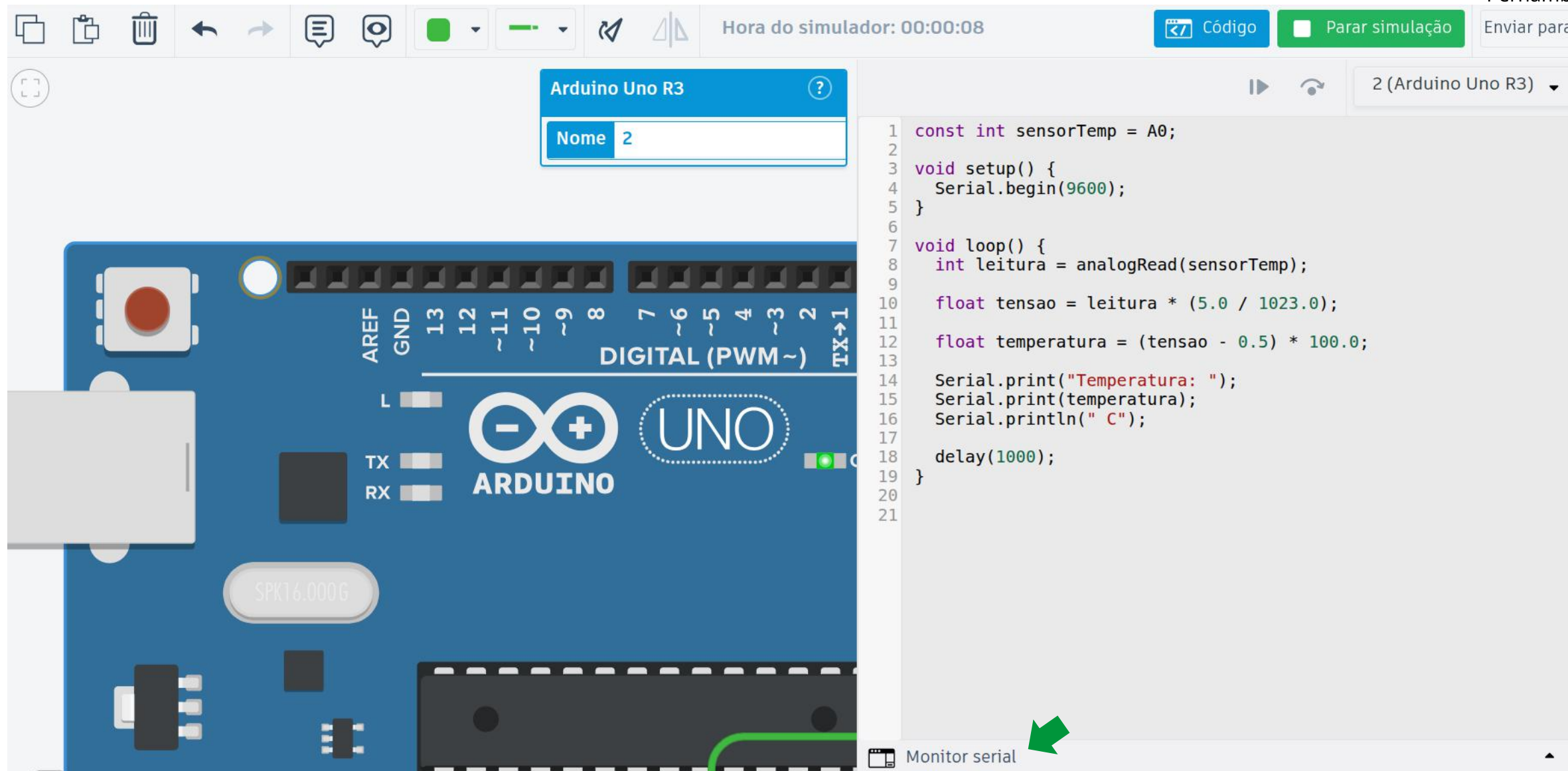
Prática: Obtendo e Exibindo a Temperatura no Monitor Serial



Prática: Obtendo e Exibindo a Temperatura no Monitor Serial

```
1  const int sensorTemp = A0;
2
3  void setup() {
4      Serial.begin(9600);
5  }
6
7  void loop() {
8      int leitura = analogRead(sensorTemp);
9
10     float tensao = leitura * (5.0 / 1023.0);
11
12     float temperatura = (tensao - 0.5) * 100.0;
13
14     Serial.print("Temperatura: ");
15     Serial.print(temperatura);
16     Serial.println(" C");
17
18     delay(1000);
19 }
```

Prática: Obtendo e Exibindo a Temperatura no Monitor Serial

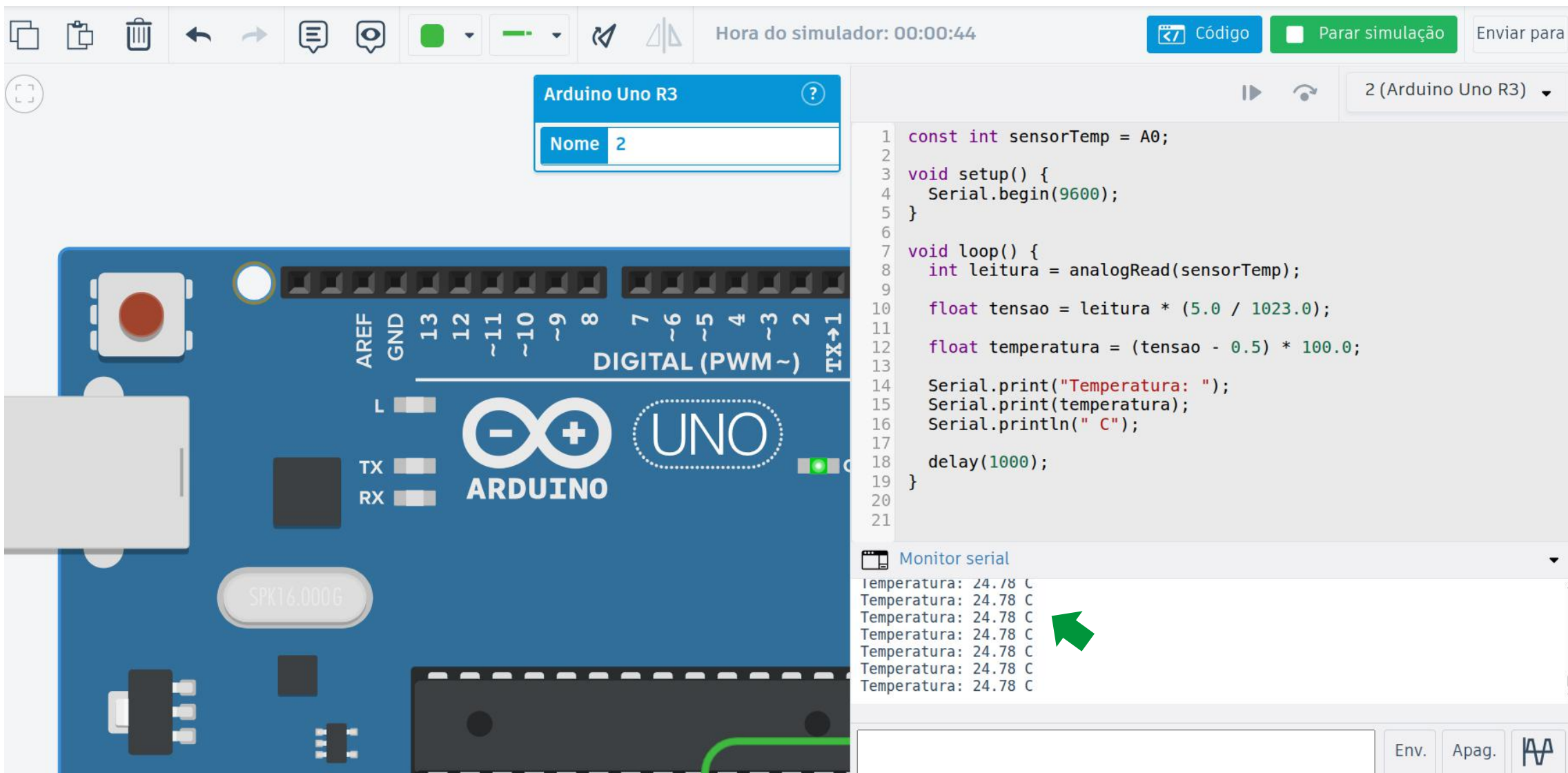


The screenshot displays the Arduino IDE environment. At the top, a toolbar contains icons for file operations, simulation control, and communication. The status bar indicates 'Hora do simulador: 00:00:08'. A dropdown menu for the board type shows 'Arduino Uno R3' with a 'Nome' field set to '2'. The main workspace shows a detailed 3D model of an Arduino Uno R3 board. To the right, the code editor contains the following C++ code:

```
1  const int sensorTemp = A0;
2
3  void setup() {
4      Serial.begin(9600);
5  }
6
7  void loop() {
8      int leitura = analogRead(sensorTemp);
9
10     float tensao = leitura * (5.0 / 1023.0);
11
12     float temperatura = (tensao - 0.5) * 100.0;
13
14     Serial.print("Temperatura: ");
15     Serial.print(temperatura);
16     Serial.println(" C");
17
18     delay(1000);
19 }
20
21
```

At the bottom right, a 'Monitor serial' window is visible, indicated by a green arrow. The window title is '2 (Arduino Uno R3)'.

Prática: Obtendo e Exibindo a Temperatura no Monitor Serial



The screenshot displays the Arduino IDE environment. On the left, a virtual Arduino Uno R3 board is shown with a digital display and a red push button. A pop-up window for the board configuration shows the name '2'. The top toolbar includes icons for file operations, simulation control, and a status bar indicating 'Hora do simulador: 00:00:44'. The main editor on the right contains the following C++ code:

```
1 const int sensorTemp = A0;
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8   int leitura = analogRead(sensorTemp);
9
10  float tensao = leitura * (5.0 / 1023.0);
11
12  float temperatura = (tensao - 0.5) * 100.0;
13
14  Serial.print("Temperatura: ");
15  Serial.print(temperatura);
16  Serial.println(" C");
17
18  delay(1000);
19 }
20
21
```

Below the code editor is the 'Monitor serial' window, which displays the output of the program:

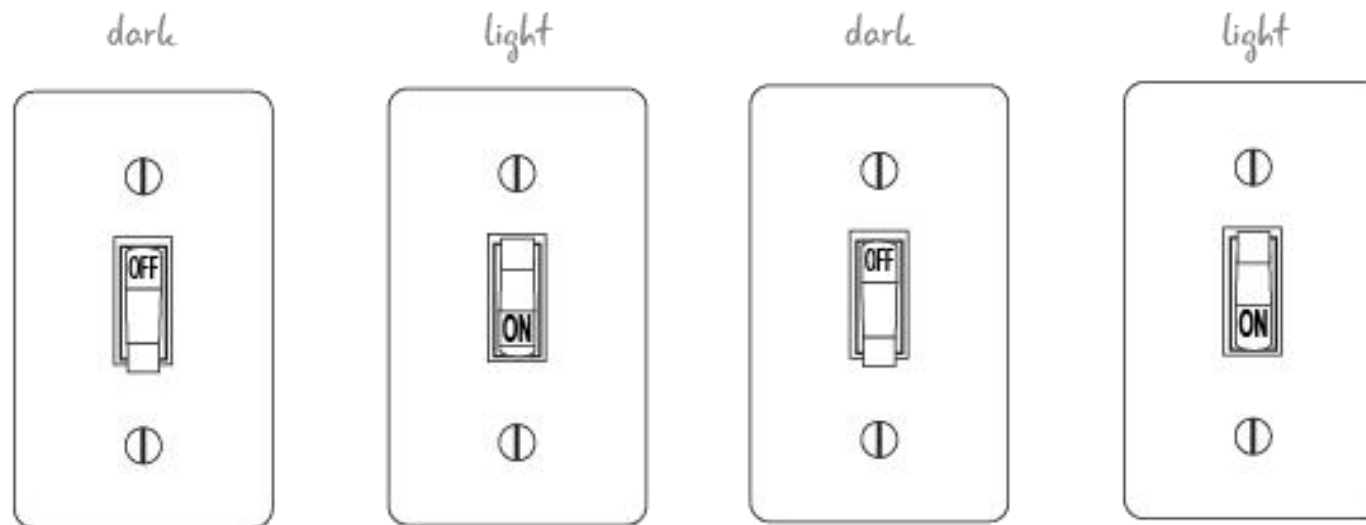
```
Temperatura: 24.78 C
Temperatura: 24.78 C
Temperatura: 24.78 C
Temperatura: 24.78 C
Temperatura: 24.78 C
Temperatura: 24.78 C
Temperatura: 24.78 C
```

A green arrow points to the serial monitor output. The bottom of the IDE features buttons for 'Env.' (Environment), 'Apag.' (Erase), and a waveform icon.

Simulando uma Saída Analógica com PWM

- O Arduino só pode gerar diretamente dois níveis de tensão em seus pinos digitais: 0 V (LOW) e 5 V (ou 3,3 V, dependendo da placa). Se ele trabalha apenas com esses valores discretos, como é possível simular uma saída analógica?
- A resposta está na técnica chamada PWM (Pulse Width Modulation - Modulação por Largura de Pulso).
- Essa técnica permite controlar o brilho de LEDs, a velocidade de motores e outros dispositivos, simulando uma saída analógica mesmo utilizando um pino digital.

Simulando uma Saída Analógica com PWM

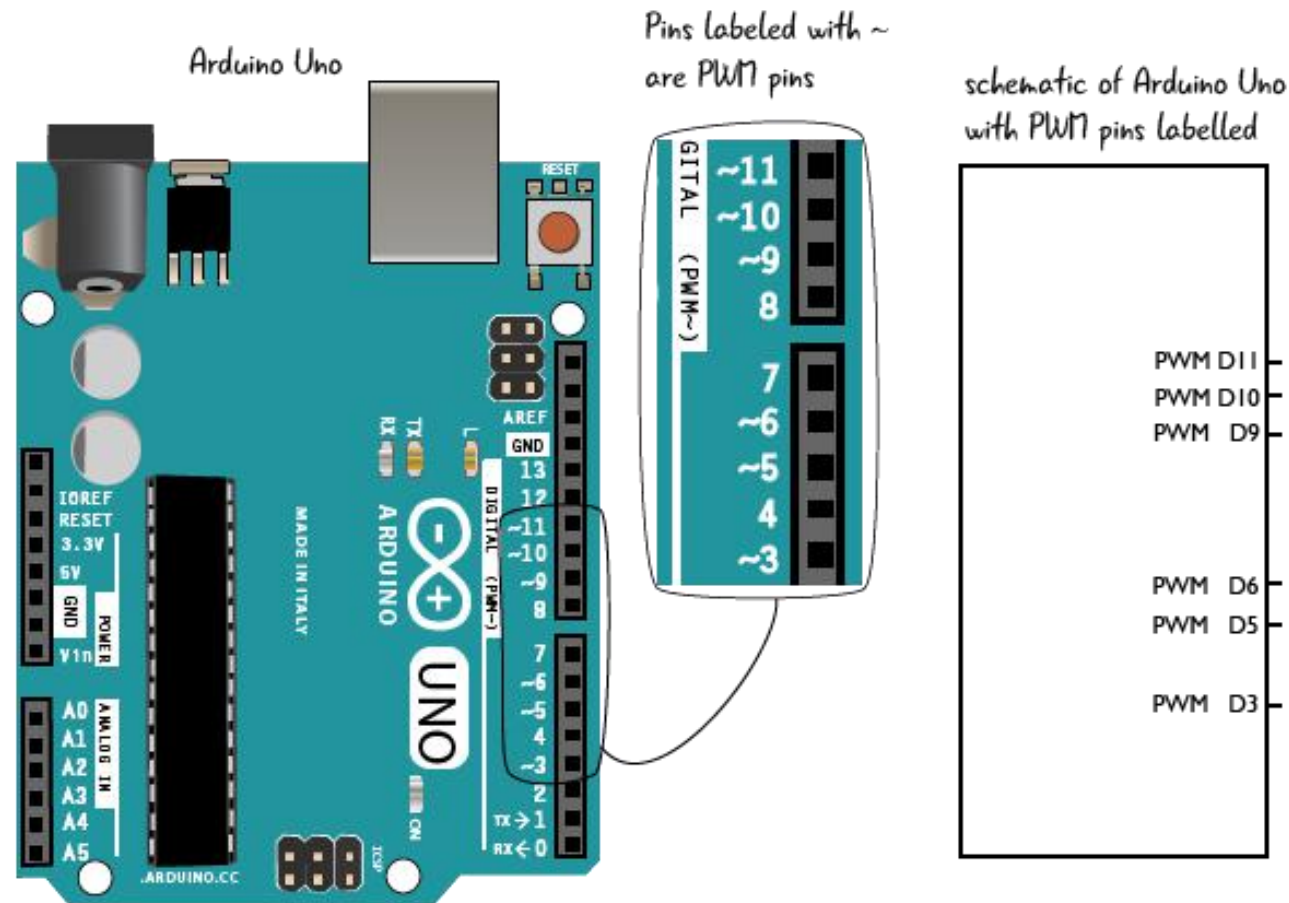


Simulando uma Saída Analógica com PWM

- PWM (Pulse Width Modulation - Modulação por Largura de Pulso) consiste em ligar e desligar o pino muito rapidamente. Ao variar a proporção de tempo em que o sinal permanece ligado em relação ao tempo em que permanece desligado (ciclo de trabalho ou duty cycle), obtém-se uma tensão média percebida pelo circuito.
- Por exemplo:
 - 0% de duty cycle $\rightarrow$ 0 V (sempre desligado);
 - 50% de duty cycle $\rightarrow$ aproximadamente 2,5 V de valor médio;
 - 100% de duty cycle $\rightarrow$ 5 V (sempre ligado).

Simulando uma Saída Analógica com PWM

- Quais pinos suportam PWM?



Simulando uma Saída Analógica com PWM

- Escrever um valor analógico com PWM:
 - Usa-se a função `analogWrite()` em pinos digitais que suportam PWM (~).

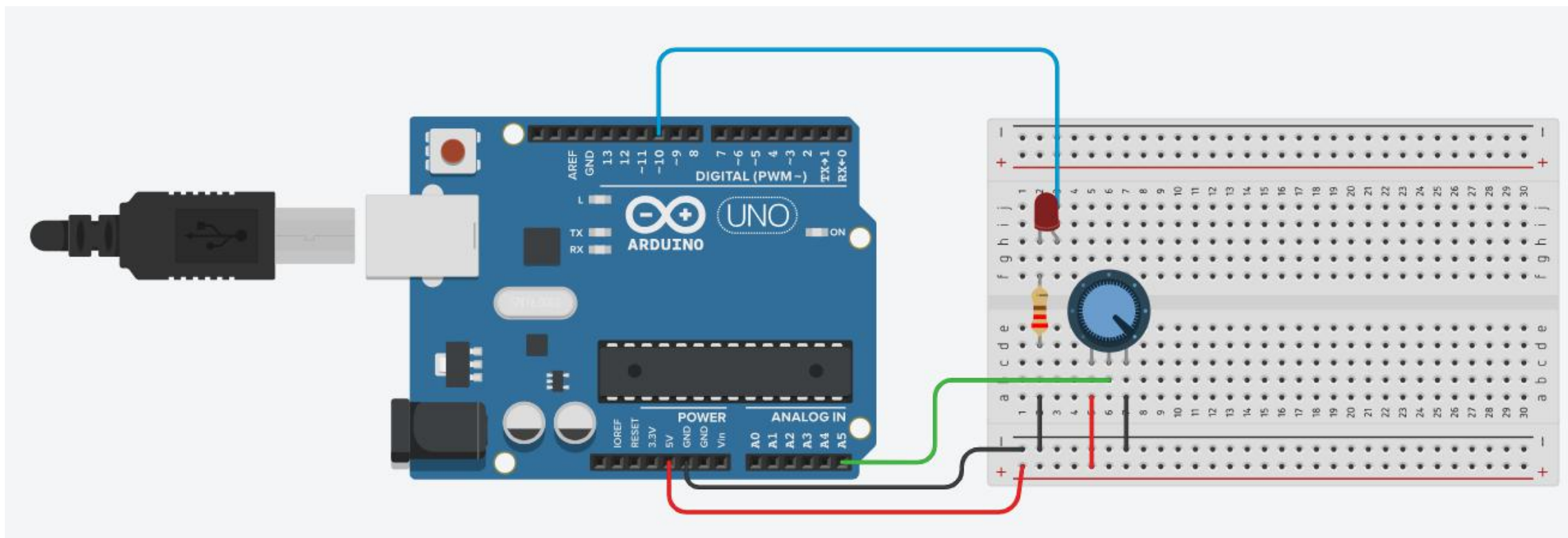
- Configuração:

```
void setup() {  
    pinMode(pin, OUTPUT);  
}
```

- Loop:

```
void loop() {  
    analogWrite(9, 128); // 50% de intensidade (aprox.)  
    delay(1000);  
    analogWrite(9, 255); // máximo  
    delay(1000);  
}
```

Prática: potenciômetro controlando o brilho de um LED



Prática: potenciômetro controlando o brilho de um LED

```
1  const int pinPotenciometro = A5;
2  const int pinLed = 10;
3
4  void setup() {
5      pinMode(pinLed, OUTPUT);
6  }
7
8  void loop() {
9      int valor = analogRead(pinPotenciometro); // 0 a 1023
10
11      int brilho = map(valor, 0, 1023, 0, 255);
12
13      analogWrite(pinLed, brilho);
14
15      delay(10);
16  }
```

Comparação entre as funções digitais e analógicas

Nome da função	O que ela faz	Parâmetros requeridos	Intervalo de valores
digitalRead()	Lê o valor de um pino de entrada digital.	O número do pino que queremos fazer a leitura.	Lê 1 ou 0 a partir do pino.
digitalWrite()	Escreve um valor para um pino de saída digital.	O número do pino e o valor a ser escrito nesse pino.	Escreve 1 ou 0 no pino.
analogRead()	Lê o valor de um pino de entrada analógica.	O número do pino que queremos fazer a leitura.	Lê um inteiro entre 0 e 1023 do pino.
analogWrite()	Escreve um valor para um pino de saída PWM.	O número do pino e o valor a ser escrito nesse pino.	Escreve um inteiro entre 0 e 255 no pino, o que resulta numa tensão entre 0V e 5V.

Desafio

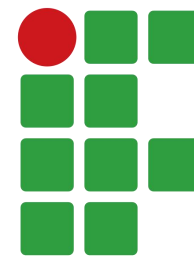
1. Conecte um sensor de temperatura TMP36 a um display de cristal líquido (LCD) no Tinkercad e desenvolva um sistema que permita a leitura da temperatura em tempo real. O circuito deve ser montado com o Arduino Uno, o sensor TMP36 e um display LCD 16x2 ou LCD 16x2 (I2C). Após a montagem, implemente um programa capaz de ler o valor analógico fornecido pelo sensor, converter essa leitura em temperatura em graus Celsius e exibir o resultado diretamente no display.

Dúvidas



Eletricidade e Eletrônica

Curso Técnico em Informática para Internet
Lucas Sampaio Leite



**INSTITUTO
FEDERAL**
Pernambuco