

Objetivo

- Utilizar as portas analógicas de entrada e saída do Arduino

Conteúdo

A placa Arduino UNO possui 6 pinos de entradas analógicas (**pinos A0 - A5**). Esses pinos são designados para a leitura de valores analógicos, como a tensão de um sensor. Os valores a serem lidos devem estar na faixa de 0 a 5 Volts, sendo convertidos em valores entre 0 e 1023.



Figura 01 – Entradas e saídas analógicas

O Arduino disponibiliza bibliotecas que simplificam o acesso aos pinos analógicos, tornando possível utilizar cada pino como entrada ou saída com o auxílio das seguintes funções:

- pinMode():** Esta função é usada para configurar um pino como entrada ou saída analógica e, geralmente, é empregada dentro da função `setup()`. Sua sintaxe é a seguinte: `pinMode(pino, modo)`.

Parâmetros:

 - pino:** O número correspondente ao pino que deseja configurar, que pode variar de A0 a A5 no caso da placa Arduino UNO.
 - modo:** Indica o modo desejado para a configuração do pino, podendo ser INPUT ou OUTPUT.
- analogRead():** Essa função lê o valor de um pino analógico. Internamente, o Arduino possui um conversor Analógico/Digital de 10 bits, resultando em um valor de retorno na faixa de 0 a 1023. Sua sintaxe é a seguinte: `int analogRead(pino)`.

Parâmetro:

 - pino:** número correspondente ao pino do qual você deseja ler um valor analógico, que, no caso da Arduino UNO, varia de 0 a 5.

Resumo dos tipos e funções disponíveis no Arduino.

Categoria	Funções
Entradas e Saídas Digitais	<code>digitalRead()</code> , <code>digitalWrite()</code> , <code>pinMode()</code>
Entradas e Saídas Analógicas	<code>analogRead()</code> , <code>analogReference()</code> , <code>analogWrite()</code>
Funções Temporizadoras	<code>delay()</code> , <code>delayMicroseconds()</code> , <code>micros()</code> , <code>millis()</code>
Comunicação	<code>Serial()</code> , <code>Stream()</code>
Números Aleatórios	<code>random()</code> , <code>randomSeed()</code>
Funções Matemáticas	<code>abs()</code> , <code>constrain()</code> , <code>map()</code> , <code>max()</code> , <code>min()</code> , <code>pow()</code> , <code>sq()</code> , <code>sqrt()</code>
Funções Trigonométricas	<code>cos()</code> , <code>sin()</code> , <code>tan()</code>

Tipos de dados
Bool, boolean, byte, char, double, float, int, long, short, size_t, string, String, unsigned char, unsigned int, unsigned long, vetor, void, word

Constantes
<code>HIGH</code> <code>LOW</code> , <code>INPUT</code> <code>OUTPUT</code> <code>INPUT_PULLUP</code> , <code>LED_BUILTIN</code> , <code>true</code> <code>false</code>

Exemplo – Sensor de temperatura

O circuito (Figura 02) e código abaixo ilustram a utilização de um sensor de temperatura e o acionamento de três LED. O simulador [Tinkercad](https://www.tinkercad.com) (<https://www.tinkercad.com>) foi utilizado na implementação do projeto.

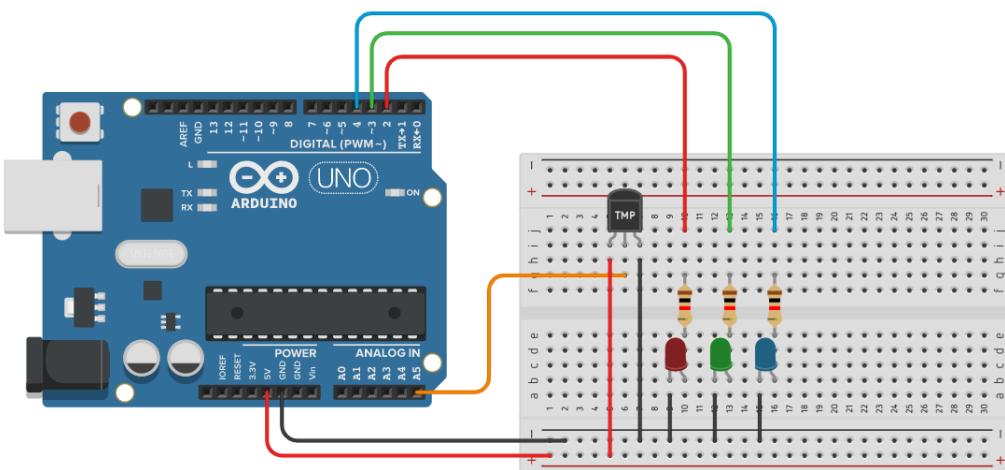


Figura 02 – Circuito da montagem do controle de temperatura

Código:

```
const int ledVermelhoPin = 2;
const int ledVerdePin = 3;
const int ledAzulPin = 4;
const int sensorTempPin = A5;

void setup(){
    pinMode(sensorTempPin, INPUT);
    pinMode(ledVermelhoPin, OUTPUT);
    pinMode(ledVerdePin, OUTPUT);
    pinMode(ledAzulPin, OUTPUT);
    Serial.begin(9600);
}

void loop(){
    float tempValor = lerTemperatura();
    Serial.println(tempValor);

    if(tempValor >= 30) {
        digitalWrite(ledVermelhoPin, HIGH);
        digitalWrite(ledVerdePin, LOW);
        digitalWrite(ledAzulPin, LOW);
    }else {
        if ((tempValor >= 22) && (tempValor < 30)){
            digitalWrite(ledVermelhoPin, LOW);
            digitalWrite(ledVerdePin, HIGH);
            digitalWrite(ledAzulPin, LOW);
        }else{
            digitalWrite(ledVermelhoPin, LOW);
            digitalWrite(ledVerdePin, LOW);
            digitalWrite(ledAzulPin, HIGH);
        }
    }
    delay(100);
}

float lerTemperatura(){
    float temp = analogRead(sensorTempPin);
    //ajuste do valor de leitura do sensor
    temp = -40 + (0.488155 * (temp - 20));
    return temp;
}
```

Exercícios

- Monte o circuito apresentado na Figura 03 e desenvolva um código para o controle do sistema de iluminação urbana ilustrado na Figura 04.

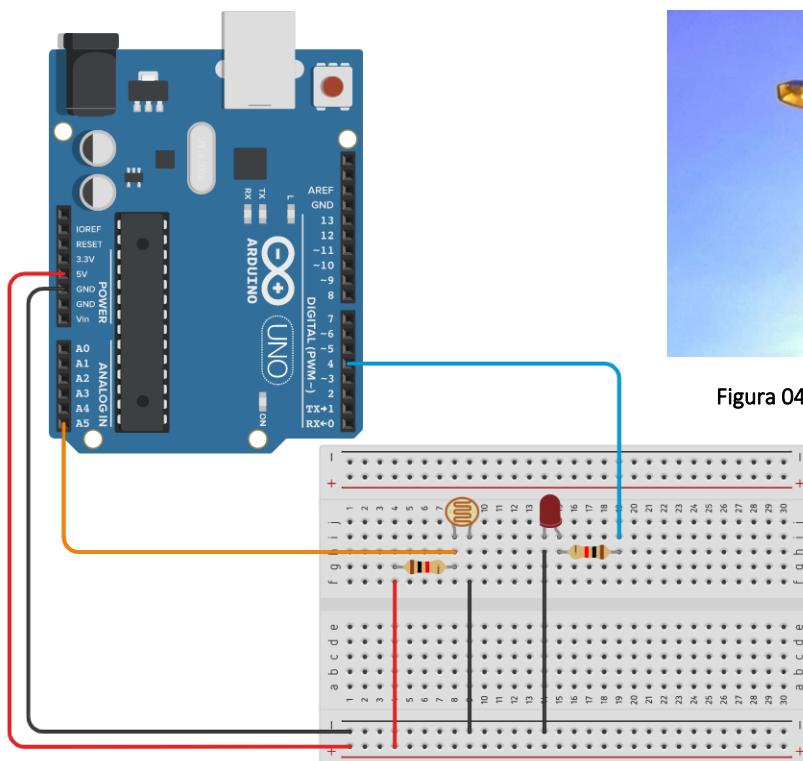


Figura 03 – Circuito de controle do sistema de iluminação



Figura 04 – Sistema de Iluminação Urbana