

Trabalhando com entradas analógicas e visualizando dados pelo Monitor Serial

As entradas digitais só podem assumir dois estados, HIGH e LOW, ou seja, 0 V ou 5 V. Dessa forma só é possível ler apenas dois estados. Por exemplo, verificar se uma porta está aberta ou fechada, identificar se um botão está pressionado ou solto, etc. Com as entradas digitais você aplica em lógica discreta para controle de seus projetos, porém em muitas situações a variação das grandezas envolvidas acontece de forma analógica. Ou seja, variam continuamente em relação ao tempo e podem assumir infinitos valores dentro de uma faixa. Como exemplo a temperatura, pressão e umidade são grandezas que variam dessa forma.

O micro controlador Arduino trabalha internamente com dados digitais, portanto é necessário traduzir um sinal analógico para um valor digital. A técnica utilizada para leitura de um sinal analógico pelo Arduino é a conversão analógica digital. Essa técnica consiste em converter o sinal analógico para um valor digital, dessa forma se pode quantificar o sinal presente no pino. Esse processo é feito pelo conversor Analógico digital, ADC ou conversor A/D.

Um conversor A/D quantifica o valor analógico conforme a quantidade de bits da sua resolução. A resolução de um conversor A/D é dada pela seguinte equação:

$$\text{resolução} = V_{ref} / 2^n$$

onde:

Vref: tensão de referência do conversor A/D;

n: número de bits do conversor.

Conversor A/D do Arduino

O conversor A/D do micro controlador ATmega328 possui 10 bits de resolução, a sua tensão de entrada pode variar de 0 V até o valor de VCC e possui referência interna selecionável de 1,1 V.

IoT - Básico

Prof. Me. Anderson Vanin

Dessa forma quando está trabalhando com a referência em VCC o menor valor que pode ser lido será:

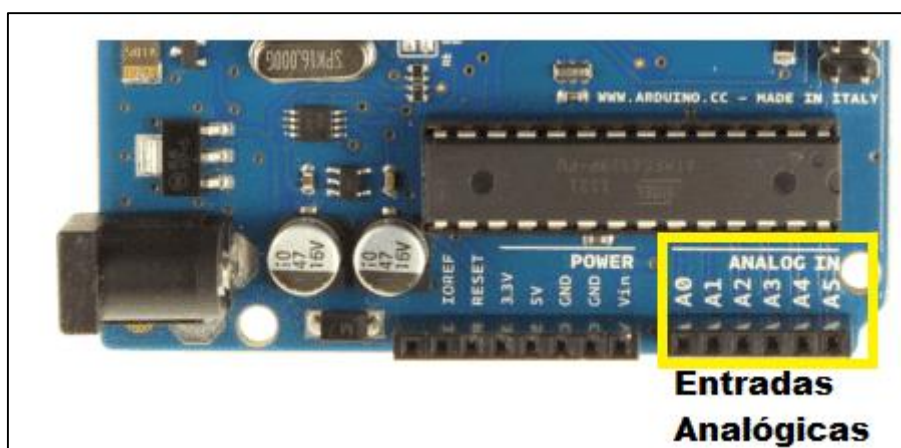
$$\text{Resolução} = 5 \text{ V} / 1024 = 4,88 \text{ mV}$$

Esse é o valor de degrau para uma conversão em 10 bits com referência em 5 V.

Caso trabalhe com a referência interna de 1,1V a resolução será:

$$\text{resolução} = \frac{1,1}{1024} = 1,07 \text{ mV}$$

A placa Arduino UNO possui 6 canais de conversor analógico digital. Essas entradas são nomeadas de A0 a A5 e são exibidas na figura a seguir:



Lê o valor presente em um pino configurado como entrada analógica. Internamente o Arduino possui um conversor A/D de 10 bits. Dessa forma o valor retornado por esta função estará na faixa de 0 a 1023 conforme o valor presente no pino.

Função map() no Arduino

A função map() permite efetuar o mapeamento de um intervalo numérico em outro intervalo numérico desejado. Isso significa que em um intervalo numérico que vai de um valor mínimo até um valor máximo, o valor mínimo será mapeado em um novo_mínimo, e o valor máximo será mapeado em um novo_máximo, e os valores intermediários serão remapeados em novos valores intermediários, de forma correspondente.

IoT - Básico

Prof. Me. Anderson Vanin

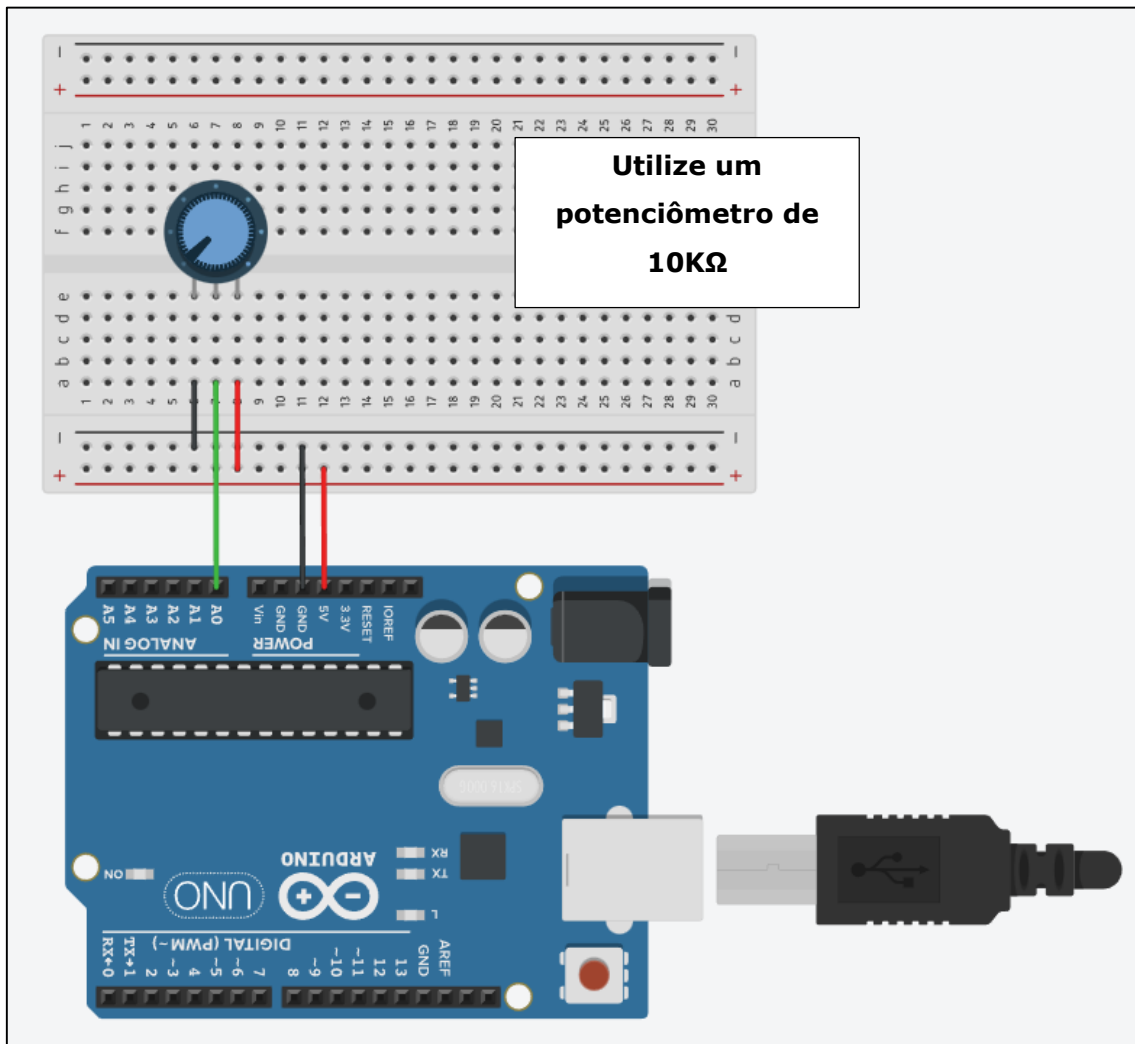
Exemplo: Queremos mapear os valores em um intervalo que vai de 0 a 1023 para um novo intervalo entre 0 e 200. Os valores atuais serão lidos a partir de uma variável chamada `valor`, e os novos valores mapeados serão gravados em uma variável de nome `novo_valor`:

```
novo_valor = map(valor,0,1023,0,200);
```

Desta forma, toda vez que um valor entre 0 e 1023 for gravado na variável `valor`, ele será mapeado para seu correspondente no intervalo entre 0 e 200, e esse novo valor será atribuído à variável `novo_valor`.

Exemplo prático Thinkercad

- Monte o circuito conforme mostrado a seguir:



- Código:

IoT - Básico
Prof. Me. Anderson Vanin

```
int sensor = A0;

void setup() {

    pinMode(sensor, INPUT);

    Serial.begin(9600);

}

void loop() {

    int valor = analogRead(sensor);

    Serial.print("Valor do pino A0: ");

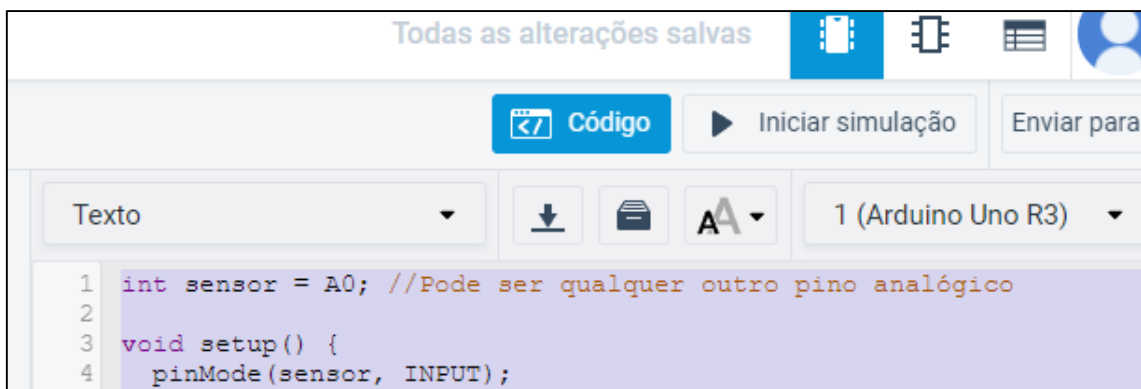
    Serial.println(valor);

    delay(1000);

}
```

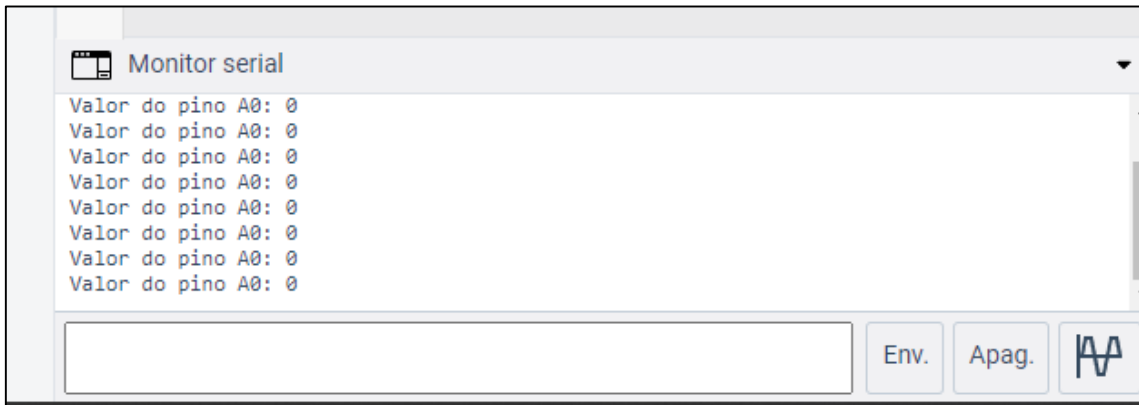
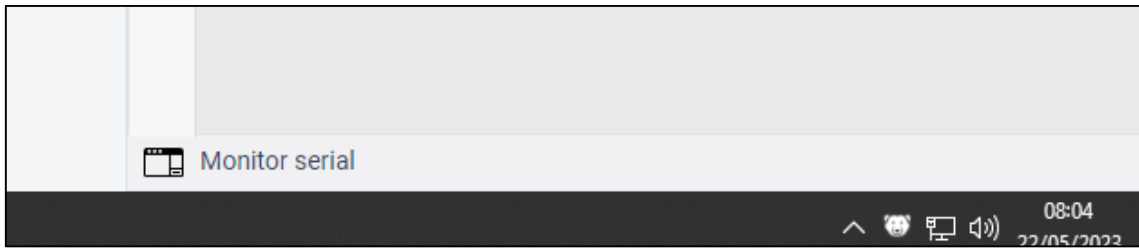
- Executando o código:

Clique para iniciar a simulação

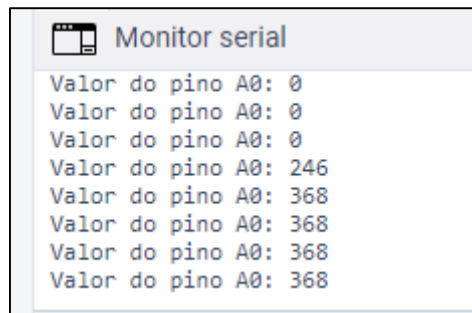


Clique para abrir o Monitor Serial na parte inferior da janela

IoT - Básico
Prof. Me. Anderson Vanin



Movimente o cursor do potenciômetro e verifique a saída no Monitor Serial



Agora suponha que você deseje ver, valores de uma porcentagem (0 a 100%). Neste caso, devemos mapear a saída ajustando estes valores que vão de 0 a 1023 para que sejam exibidos de 0 a 100.

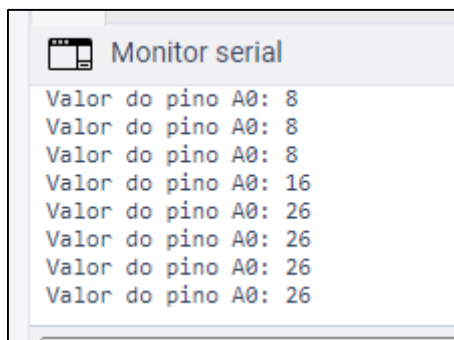
Altere o código:

```
int sensor = A0;

void setup() {
  pinMode(sensor, INPUT);
  Serial.begin(9600);
}
```

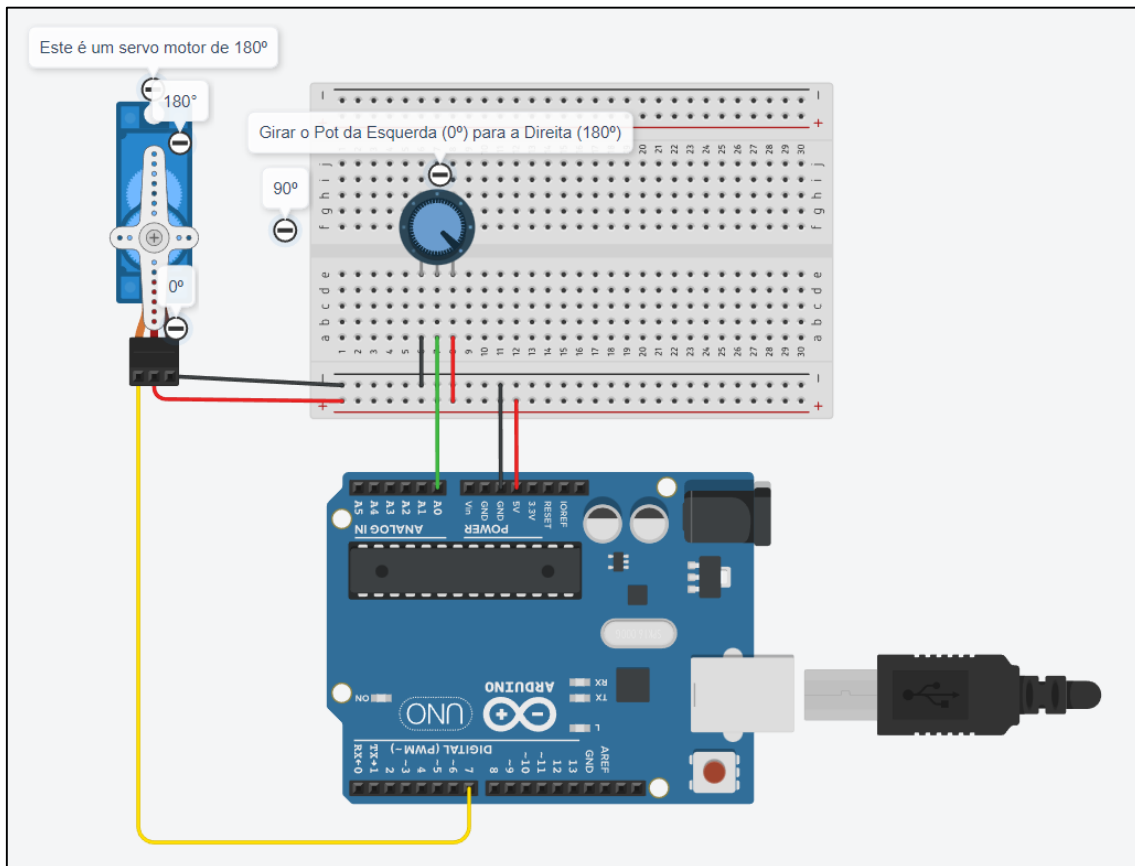
IoT - Básico
Prof. Me. Anderson Vanin

```
void loop() {  
  
    int valor = analogRead(sensor);  
  
    int valor_mapeado = map(valor,0,1023,0,100);  
  
    Serial.print("Valor do pino A0: ");  
  
    Serial.println(valor_mapeado);  
  
    delay(1000);  
  
}
```



Repare que agora as leituras mostradas estão configuradas para uma escala de 0 a 100.

Exemplo com Servo Motor – Posicional de 0° a 180°



```
#include <Servo.h>
```

```
int sensor = A0;
```

```
int posicao = 0;
```

```
Servo servo;
```

```
void setup() {
```

```
    pinMode(sensor, INPUT);
```

```
    Serial.begin(9600);
```

```
    servo.attach(7);
```

```
    servo.write(0); // INICIA O MOTOR NA POSIÇÃO 0°
```

```
}
```

IoT - Básico
Prof. Me. Anderson Vanin

```
void loop() {  
  
    int valor = analogRead(sensor);  
  
    int valor_mapeado = map(valor,0,1023,0,180);  
  
    Serial.print("Angulo: ");  
  
    Serial.print(valor_mapeado);  
  
    Serial.println(" Graus");  
  
    delay(1000);  
  
    posicao = valor_mapeado;  
  
    servo.write(posicao);  
  
    delay(5);  
  
}
```