

Conversor AD

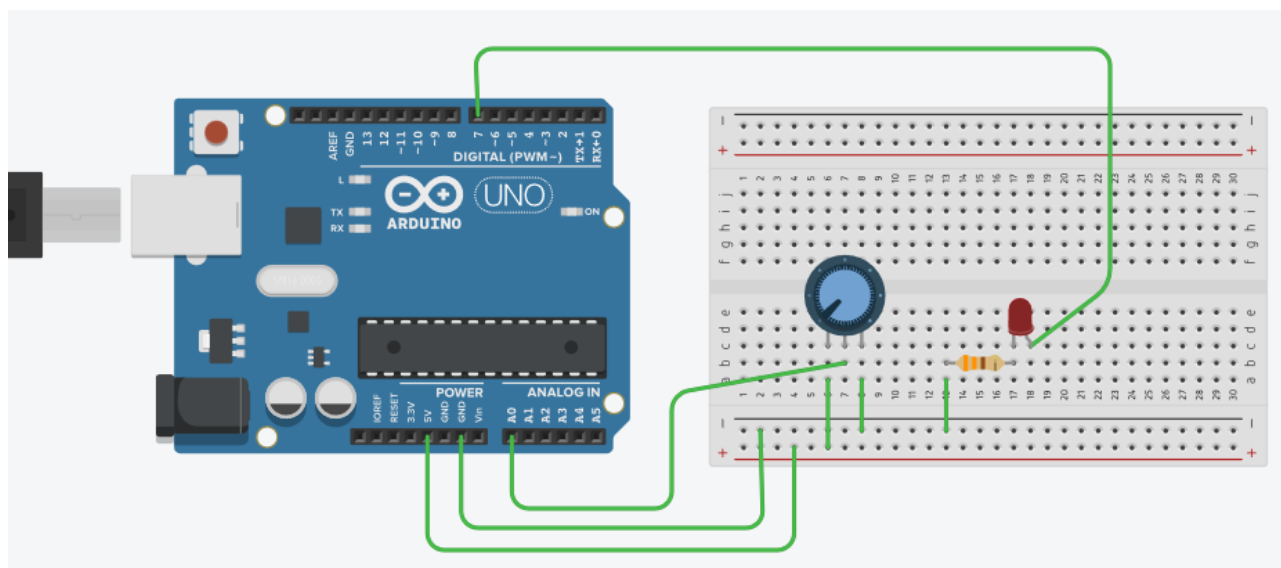
Estudamos as portas digitais do Arduino e vimos que podemos receber ou enviar sinais de 5V ou 0V através dos pinos dessas portas. Porém existem dispositivos que trabalham de forma analógica, ou seja, podem assumir qualquer valor real dentro de um intervalo de uma grandeza elétrica. Como exemplo temos sensores de temperatura, inclinação, nível, etc. Como em um sistema digital não há possibilidade de se representar um número infinito de opções, é preciso resolver este problema e para isso recorremos ao conversor analógico digital CAD (ou ADC). Na prática ele faz a correspondência de um conjunto infinito de números (analógico – real) para um número finito de opções (digital).

O Arduino tem um único AD que é compartilhado entre seus pinos começados por A e vai de A0 a A5. Este AD tem um registrador de 10bits que pode ter 1024 opções diferentes, indo de 0 a 1023. Portanto se temos um sensor que trabalha em um intervalo de 0 a 5V, teremos uma correspondência do 0V com 0 e 1023 em 5V, sendo os valores intermediários dispostos linearmente. Temos portanto $5V/1023 = 4,89mV$ de resolução e isto indica que uma tentativa de identificar um intervalo menor que a resolução não vai ser possível. Felizmente para a maioria dos sistemas embarcados esta resolução será suficiente.

Outros parâmetros como a taxa de amostragem também não serão vistos neste material e também estas especificações serão suficientes para nossos propósitos.

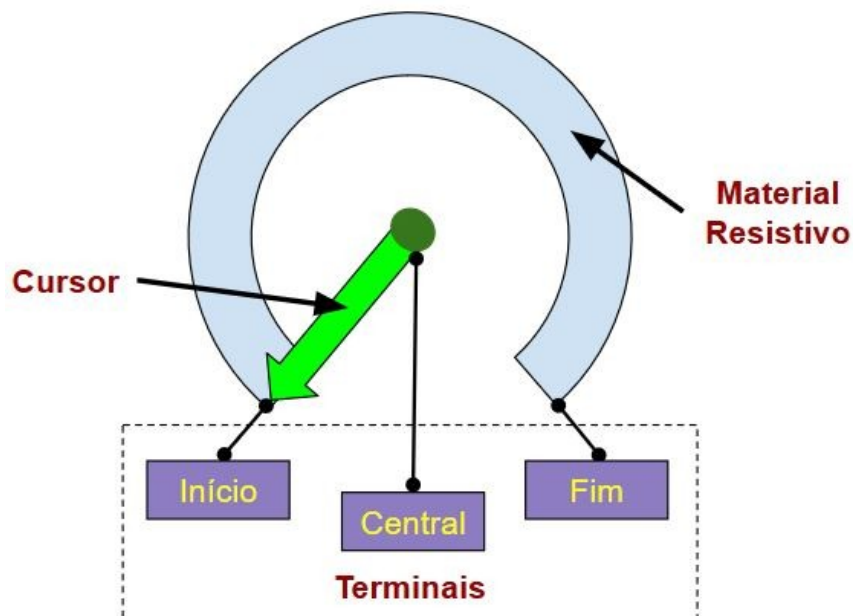
Exemplo 1

No circuito abaixo, usaremos o potenciômetro para fazer um LED piscar mais rápido ou mais lento de acordo com o giro do cursor do potenciômetro.



Circuito com LED e potenciômetro

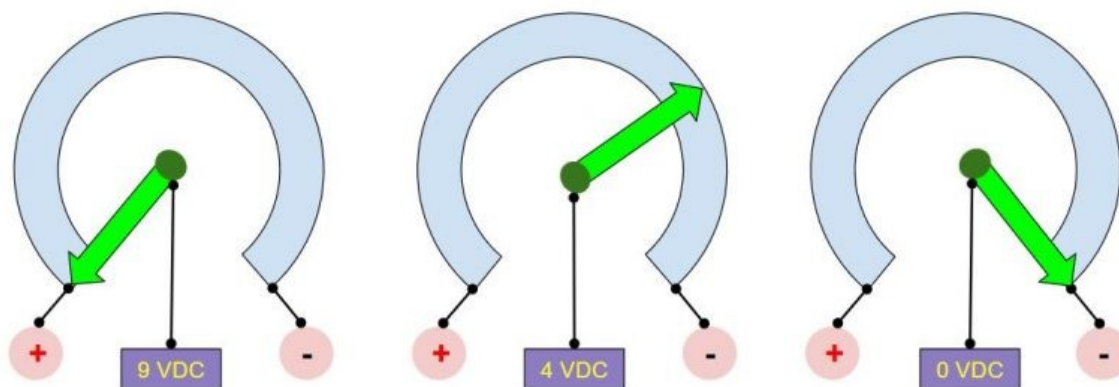
Lembrando que um potenciômetro possui três terminais. Um par externo (terminais “*início*” e “*fim*”) se conecta com os lados opostos de um elemento resistivo interno, chamado de *trilha* ou *pista*. O terceiro terminal (central) se conecta internamente com um contato chamado de *cursor*, que fica em contato com a pista e pode ser movido de um extremo dessa pista ao outro, por meio da rotação de um eixo ou ainda movendo-se um controle deslizante.



Partes de um potenciômetro – fonte: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-um-potenciometro/>

Quando aplicamos uma tensão elétrica entre os terminais de início e fim, a tensão elétrica obtida no terminal central irá variar conforme o cursor se move ao longo da pista. Assim, o potenciômetro funciona como um divisor de tensão resistivo. Sua ideia é obtermos diferentes níveis de tensão elétrica, e não diferentes níveis de resistência – esse é o papel de um tipo especial de potenciômetro, o reostato, do qual falaremos mais adiante.

A figura a seguir ilustra valores de tensão obtidos em um potenciômetro ligado a uma bateria de 9V com o cursor nas posições início, aleatória qualquer e de fim:



Funcionamento de um potenciômetro como divisor de tensão.

No nosso caso, como a alimentação do Arduino é de 5V, colocaremos 5V e GND nos terminais externos e o cursor dará um valor entre 0 e 5V e o AD mapeará para valores entre 0 e 1023.

Programa:

```
#define LED 7
#define POT A0

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

void loop(){
    int tempo;
    tempo = analogRead (POT);

    digitalWrite(LED, HIGH);
    delay (tempo+100);
    digitalWrite(LED, LOW);
    delay (tempo+100);
}
```

Analisando o programa, vemos que temos os #defines para indicar os pinos ligados ao LED e ao cursor do POT, na função setup() vemos o pino do LED como saída. Note que **não devemos indicar os pinos analógicos como entrada**. Somente precisamos fazer isso com os pinos digitais.

Na função loop() temos a função analogRead() que atribui um valor de 0 a 1023 à variável tempo.

```
tempo = analogRead (POT);
```

Esta variável será usada como argumento da função delay() acrescida de 100. Portanto teremos um intervalo de 100ms a 1123ms para o LED de acordo com o giro do cursor do potenciômetro.

Exemplo 2:

Usando o mesmo circuito do Exemplo 1, utilizaremos outro programa que faz com que o LED acenda se o potenciômetro estiver com o cursor após a metade do seu curso e apague se estiver antes da metade.

Programa:

```
#define LED 7
#define POT A0

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

void loop(){
    if (analogRead(POT) < 512)
        digitalWrite (LED, LOW);
    else
        digitalWrite (LED, HIGH);
}
```