

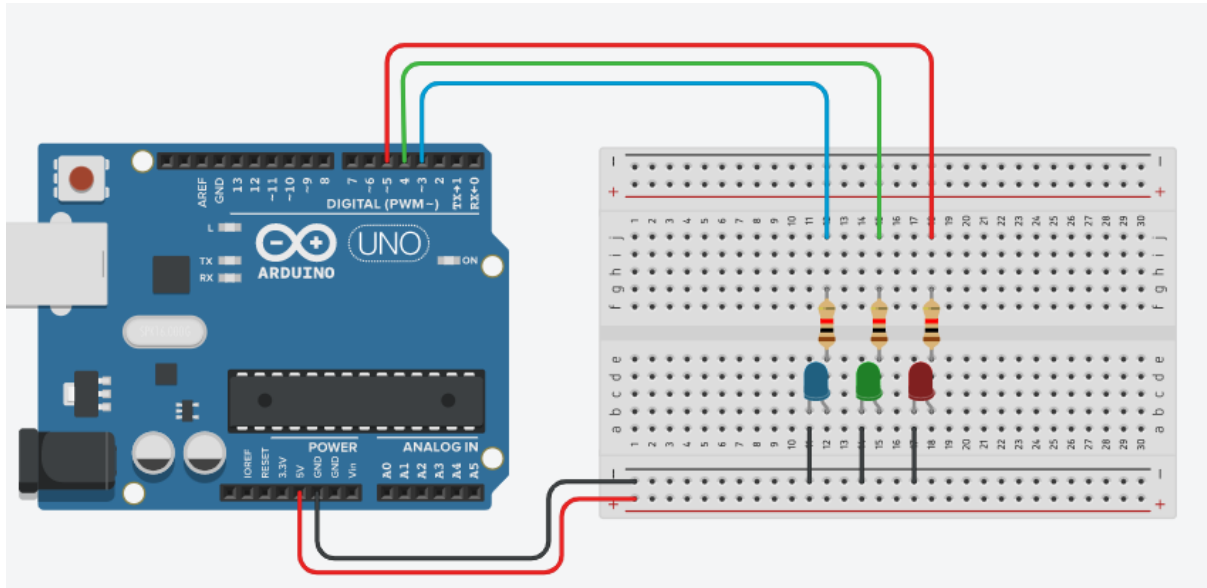
Introdução à Programação em Arduino

Ex Machina

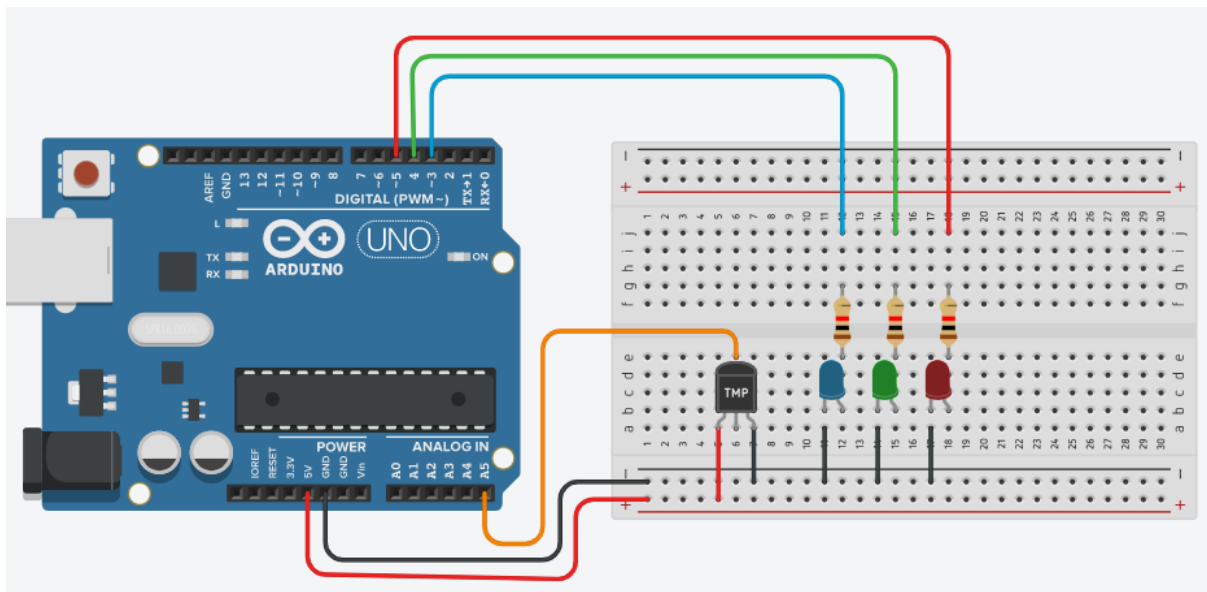
Exercícios:

1) Para este exercício, serão necessárias as funções `pinMode`, `delay`, `digitalWrite`, `analogRead` e `map`.

a) Faça os LEDs do circuito a seguir, disponível [neste link](#), acenderem um por vez a cada 1 segundo.



b) Então adicione um sensor de temperatura conectado a um pino analógico, como na imagem a seguir. O pino analógico conectado irá ler “20” quando a temperatura for a menor possível, e “358” quando for a maior. Faça o LED azul acender quando a temperatura for menor que 10, o verde quando a temperatura for maior ou igual a 10 e menor que 30, e o vermelho quando a temperatura for maior ou igual a 30.



Utilização de servo motores

A biblioteca “Servo.h”, assim como foi apresentado nos slides, pode ser usada para auxiliar no controle dos servo motores. O link de referência para essa biblioteca é o seguinte:

<https://www.arduino.cc/reference/en/libraries/servo/>

Exercícios:

- 1) Explique como funcionam os métodos attach(), write() e read() da biblioteca servo.h
- 2) Crie o circuito apresentado na imagem abaixo no tinkercad e faça um código para controlar o servo motor de acordo com o valor do potenciômetro. O potenciômetro possui valores de 0 a 1023 e o servo motor valores de 0 a 180.

Obs.: Para o correto funcionamento é preciso mapear o valor obtido no potenciômetro para que o circuito funcione corretamente. Isso pode ser feito utilizando uma das funções que foi apresentada no slide.

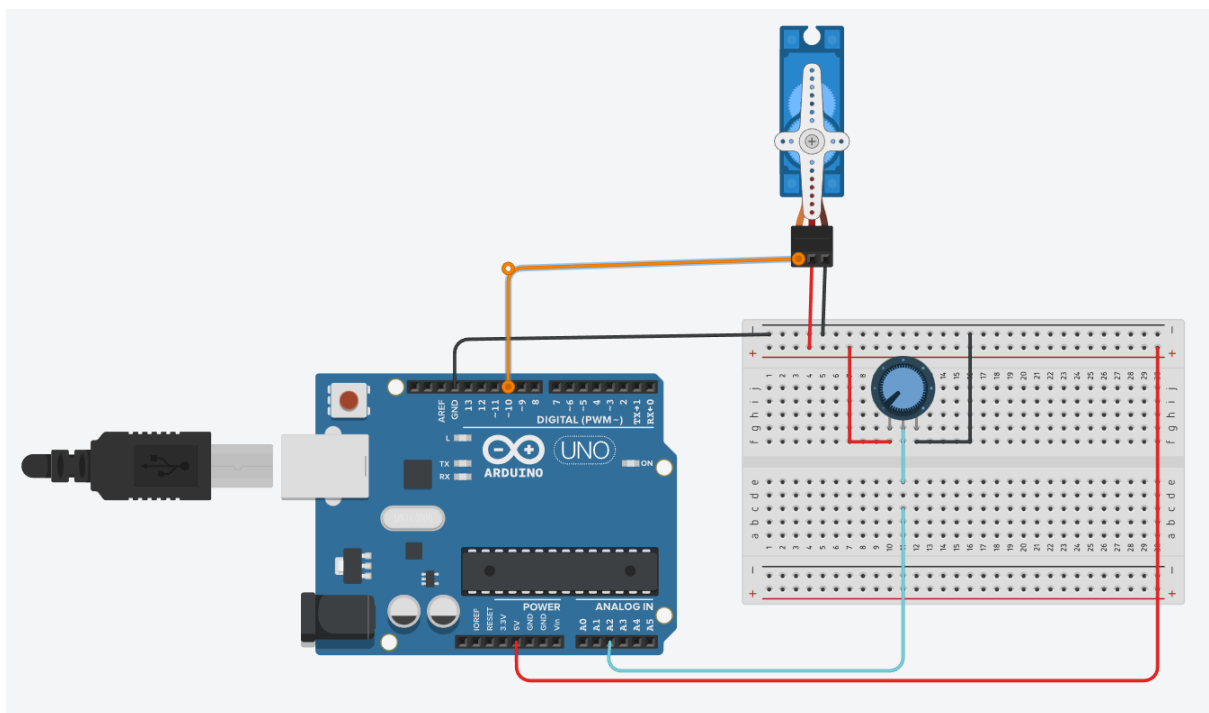


Imagem 1 - Circuito do exercício 2

- 3) Crie um circuito apenas com o servo motor e faça um código para controlar o servo motor de acordo com a palavra recebida pelo Monitor Serial (no

Tinkercad o Monitor Serial fica na parte inferior de onde é escrito o código). A tabela 1 apresenta cada palavra com o seu respectivo movimento.

Palavra	Movimento
“hor”	O servo deverá realizar um movimento de 180° suave no sentido horário
“ant”	O servo deverá realizar um movimento de 180° suave no sentido anti-horário
Qualquer outra palavra	Escolha um valor fixo para o servo

Tabela 1 - Lista de palavras e movimentos

Obs.: Para esse exercício é necessário esperar ser recebido algum valor no Monitor Serial e somente depois realizar o respectivo movimento.

- 4) A prótese do projeto utiliza servo motores para realizar o movimento dos dedos. Pensando nisso, neste exercício iremos simular o funcionamento da prótese através do tinkercad. Para isso, é preciso montar um circuito utilizando 5 servo motores, cada um sendo conectado a um pino digital (lembre-se de conectar em pinos digitais PWM do arduino - Pinos 11, 10, 9, 6, 5, 3).

Depois que o circuito estiver montado, crie uma função que recebe 5 valores inteiros, um para cada servo (que controla um dedo da prótese).

A função desenvolvida deverá ser usada na função loop para simular alguma posição dos dedos que represente algum gesto (por exemplo, o hang loose).

Explicação do funcionamento do servo motor na prótese: para a realização dos movimentos na prótese usamos o valor mínimo do servo motor para representar os dedos que estão abertos e o valor máximo para os dedos que devem ser fechados.

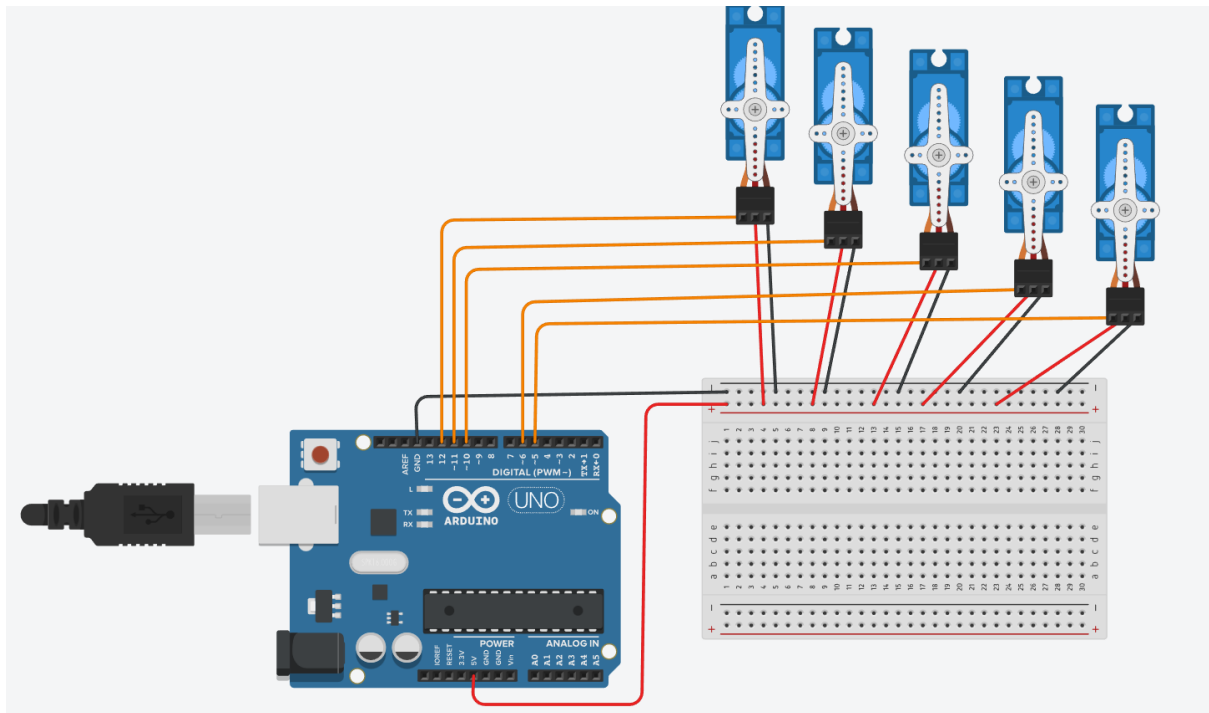


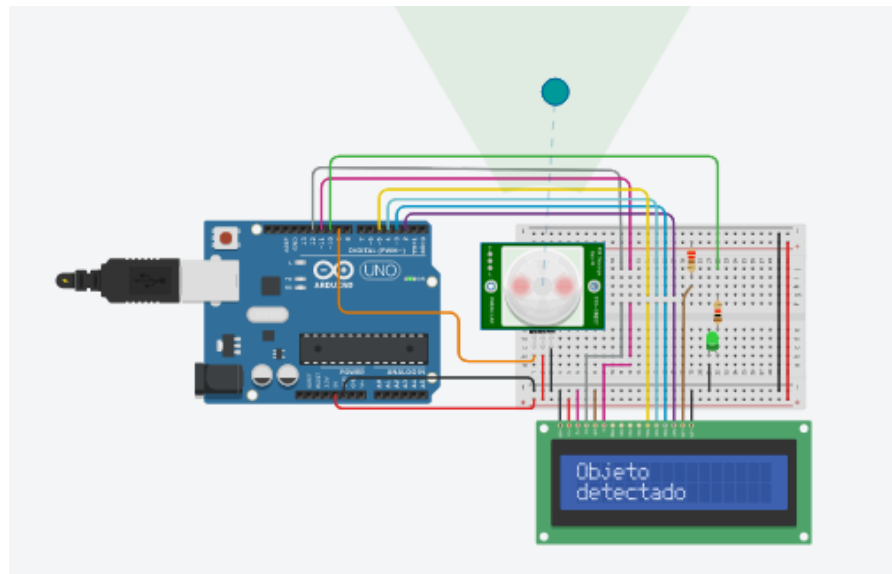
Imagem 2 - Circuito do exercício 4

Utilização de Comunicação Serial e LCD

- 5) Para este exercício será necessário incluir a biblioteca `LiquidCrystal.h` e inicializa-la [`LiquidCrystal lcd(12, 11, 5, 4, 3, 2);` e `lcd.begin(16, 2);`], e utilizar as funções `pinMode()` e `digitalWrite()`.

Sensor de Presença

Crie uma cópia do [circuito](#) e implemente um código para que ele funcione como um sensor de presença. Caso detecte presença, acione o led e escreva “Objeto detectado” no display lcd, caso contrário, desligue o led e escreva “Objeto não detectado” no display lcd, como mostra na figura abaixo:



Exercício Extra de Arduino

Ex Machina

Dado o projeto disponível [neste link](#), faça com que o display de LCD deixe de ficar com o efeito de *flicker*, ou seja, que ele atualize suavemente para outra mensagem **apenas quando necessário**. Assuma que para que o projeto cumpra sua função, não é possível acrescentar mais tempo de *delay* na função loop.