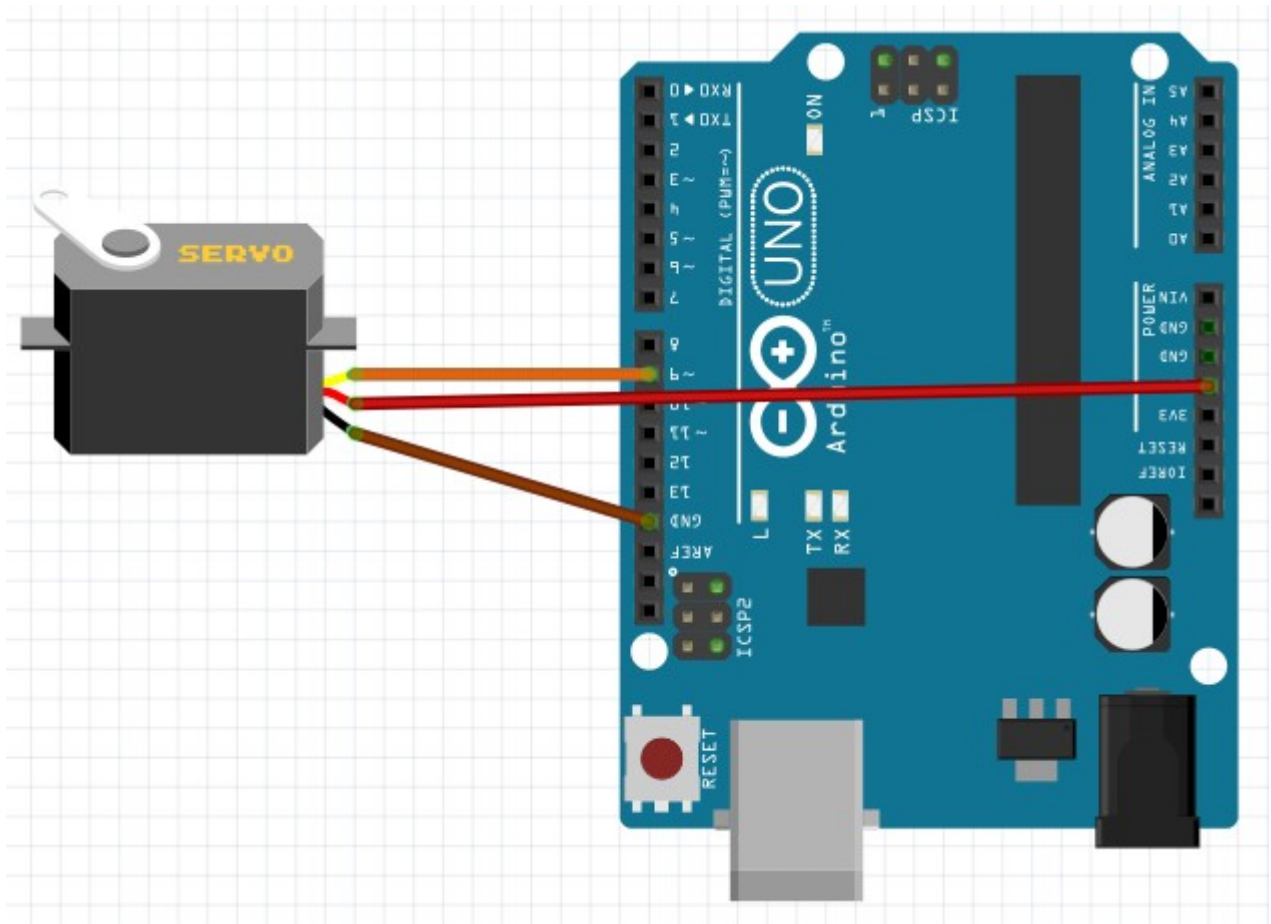


Servo mão

Vamos fazer um ‘arremessador’ utilizando um servo motor. O objetivo é que um pequeno objeto, por exemplo uma bolinha de papel, colocado na ‘mão’ seja arremessado.

Você precisará de um servo motor, três fios para conexão à placa do Arduino e, claro, do Arduino. Ficará assim:



O fio vermelho é ligado no 5V; o fio laranja é ligado no pino 9 e o pino marrom (ou preto, conforme o modelo) é ligado no GND.

Primeiramente, vamos executar o código que segue para descobirmos as posições de 0° (zero graus) e 90° (noventa graus) do nosso servo motor. O objetivo é prendermos a ‘mão’ de forma que na posição de 0° ela possa receber a bolinha, que ela via arremessar para a posição de 90° (ou outra se você trocar o código).

Código 1 – Posições 0° e 90°

```
#include <Servo.h>
Servo mao;
#define pinoServo 9
void setup() {
    mao.attach(pinoServo);
    Serial.begin(9600);
}
```

```

void loop() {
  mao.write(0);
  Serial.println("Esta na posicao zero");
  delay(1500);
  mao.write(90);
  Serial.println("Esta na posicao noventa");
  delay(1500);
}

```

Comentários:

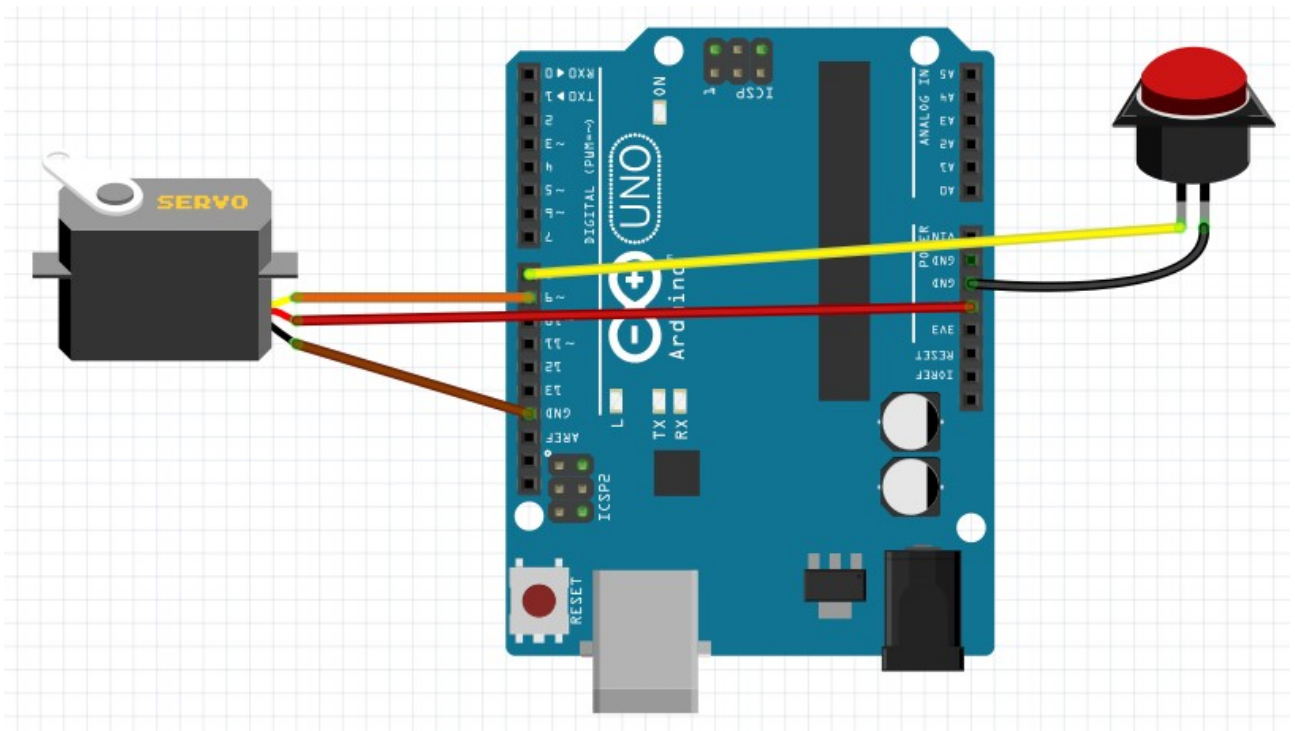
- Foi incluída a biblioteca 'Servo.h', a qual contém os códigos de controle do servo motor (`#include <Servo.h>`)
- Foi declarado um objeto que usa a biblioteca Servo, com o nome 'mao' (`Servo mao;`)
- Foi definido que o pino por meio do qual o servo motor será controlado é o '9'; se você usou outro pino, atualize o código para que o controle funcione (`#define` e que será identificado pelo nome 'pinoServo' `pinoServo 9`)
- Na função '`void setup()`', foi informada a conexão do servo para o pino escolhido, por meio da declaração '`mao.attach(pinoServo);`'
- Na função '`void setup()`', foi inicializada a biblioteca de comunicação serial com velocidade de 9600 bps (`Serial.begin(9600);`). Isto será utilizado se quisermos ligar o monitor serial para ver a informação de posição do servo
- Na função '`void loop()`', o servo é deslocado para a posição de 0° (`mao.write(0);`)
- Após o deslocamento do servo, a informação de sua posição é escrita na interface serial (`Serial.println("Esta na posicao zero");`)
- Após, há uma espera de 1,5 segundos (`delay(1500);`)
- O servo é deslocado para a posição de 90° (`mao.write(90);`)
- Após o deslocamento do servo, a informação de sua posição é escrita na interface serial (`Serial.println("Esta na posicao noventa");`)
- Após, há uma espera de 1,5 segundos (`delay(1500);`)

Após verificarmos o funcionamento e as posições do servo motor podemos prender a 'mão'. No meu caso foi utilizado um garfo de plástico, que foi torcido utilizando um soprador de ar quente. Ficou assim:



Prenda a 'mão' e certifique-se de que ela poderá girar livremente. No meu caso coloquei o servo motor em cima de um pequeno apoio, preso com fita dupla face. Depois, digite o código 2, a seguir. Você precisará, além do servo motor, de um interruptor.

A ligação ficará assim:



Caso você não possua um interruptor no qual os fios possam ser ligados diretamente, utilize uma placa de experimentação (*protoboard*) como auxílio.

Código 2 – Lançamento controlado por interruptor

```
#include <Servo.h>
Servo mao;
#define pinoServo 9
#define botao 8
void setup() {
  mao.attach(pinoServo);
  Serial.begin(9600);
  pinMode(botao, INPUT_PULLUP);
}

void loop() {
  mao.write(0);
  Serial.println("Esta na posicao zero");
  if (digitalRead(botao) == LOW){
    mao.write(90);
    Serial.println("Esta na posicao noventa");
    delay(1500);
  }
}
```

Comentários (o que foi acrescentado ou modificado no código anterior está em **negrito>**):

- Foi incluída a biblioteca 'Servo.h', a qual contém os códigos de controle do servo motor (**#include <Servo.h>**)
- Foi declarado um objeto que usa a biblioteca Servo, com o nome 'mao' (**Servo mao;**)

- Foi definido que o pino por meio do qual o servo motor será controlado é o '9' e que será identificado pelo nome 'pinoServo'; se você usou outro pino, atualize o código para que o controle funcione (`#define pinoServo 9`)
- **Foi definido que o pino por meio do qual o interruptor será lido é o '8'; se você usou outro pino, atualize o código para que o interruptor funcione (`#define botao 8`)**
- Na função '`void setup()`', foi informada a conexão do servo para o pino escolhido, por meio da declaração '`mao.attach(pinoServo);`'
- Na função '`void setup()`', foi inicializada a biblioteca de comunicação serial com velocidade de 9600 bps (`Serial.begin(9600);`). Isto será utilizado se quisermos ligar o monitor serial para ver a informação de posição do servo
- Na função '`void setup()`', foi declarado que o pino no qual o interruptor foi conectado será uma entrada (`pinMode(botao, INPUT_PULLUP);`), e que ela normalmente terá nível alto (por meio do 'PULLUP') – portanto, quando for lido um nível baixo significa que o interruptor foi pressionado
- Na função '`void loop()`', o servo é deslocado para a posição de 0° (`mao.write(0);`)
- Após o deslocamento do servo, a informação de sua posição é escrita na interface serial (`Serial.println("Esta na posicao zero");`)
- ~~Após, há uma espera de 1,5 segundos (`delay(1500);`)~~ foi deslocado para dentro do 'if'
- **Foi realizado um teste (if). Se o interruptor estiver em nível baixo (if (`digitalRead(botao) == LOW` {}), então o servo será movido para a posição 90° (`mao.write(90);`), a informação de posição será atualizada na interface serial (`Serial.println("Esta na posicao noventa");`) e haverá uma espera de 1,5 segundo (`delay(1500);`)**

Agora você está pronto para fazer uma competição. Coloque objetos na servo mão e veja até que distância eles serão arremessados. Teste com diferentes comprimentos do 'braço'.



Se você quiser deixar a competição mais interessante, pode trocar os valores dos ângulos de arremesso. A cada momento em que for arremessado será um ângulo diferente, aleatório. Você pode fazer isso incluindo uma função que crie e retorne um número aleatório. Poderia ser assim:

```
int aleatorio() {  
    int sorteio = random(4, 10);  
    return (sorteio * 10);  
}
```

Comentários :

- Foi declarada uma função chamada 'aleatorio', a qual retorna um número inteiro (`int aleatorio()`)
- Foi utilizada a função 'random', a qual retorna um número aleatório, entre um mínimo, inclusive, e um máximo (`random(4, 10)`). Neste caso, foram passados os parâmetros mínimo 4 e máximo 10, então serão gerados números entre 4 e 9 (nove, um antes do máximo)
- A função aleatorio retorna o número gerado multiplicado por 10 (então, retornará um número entre 40 e 90, o qual será utilizado para posicionamento do servo motor)
- Para que o número seja realmente aleatório, o gerador 'random' é inicializado com uma 'semente' – para isso utilizaremos a sugestão padrão de leitura de uma porta analógica desconectada, o que produzirá a leitura de um ruído, o qual é aleatório. É utilizado o comando `()` inserido na função `()`

O Código 3, com ângulo aleatório, ficará assim (o que foi acrescentado ou modificado está em **negrito**):

```
#include <Servo.h>
Servo mao;
#define pinoServo 9
#define botao 8
int aleatorio() {
    int sorteio = random(4, 10);
    return (sorteio * 10);
}

void setup() {
    mao.attach(pinoServo);
    Serial.begin(9600);
    pinMode(botao, INPUT_PULLUP);
    randomSeed(analogRead(A0));
}

void loop() {
    mao.write(0);
    Serial.println("Esta na posicao zero");
    if (digitalRead(botao) == LOW){
        mao.write(aleatorio());
        Serial.println("Esta na posicao noventa");
        delay(1500);
    }
}
```

Boa diversão.