

# **MOVIMENTOS ANGULARES COMANDADOS A PARTIR DE POTENCIÔMETROS**

**AUTOR: MARCOS REGES MOTA**

**Tópicos:**

**1 – Introdução**

**2 – Objetivo**

**3 – Desenvolvimento**

**3.1 - Componentes e seus princípios de funcionamento**

**3.1.1 - Garra**

**3.1.2 - Servo Motor**

**3.1.3 - Arduino**

**3.1.4 - Potenciômetro**

**3.1.5 - Protoboard**

**3.2 - Construção e funcionamento**

**3.2.1 - Montagem e Conexão**

**3.2.2 - Programação**

**4 - Considerações finais**

**5 - Referências**

## **1. INTRODUÇÃO**

O projeto movimentos de uma garra sincronizado aos comandos de potenciômetros se desenvolve com o intuito de fornecer uma das muitas maneiras de como se controlar uma garra. Desta vez utilizando potenciômetros como base onde a cada movimento executado no mesmo a garra deverá responder de mesma magnitude.

## **2. OBJETIVO**

Este projeto é designado à estudantes e curiosos pelo assunto que anseiam por trabalhar nesta área ou só por curiosidade e que desejam também entender o funcionamento dos equipamentos principais a ser utilizados neste projeto.

## **3. DESENVOLVIMENTO**

### **3.1 Componentes e seus princípios de funcionamento**

Para a execução do nosso protótipo utilizamos os seguintes componentes e sua quantidade:

Garra - 1 unidade

Servo Motor - 4 unidades

Arduino - 1 unidade

Potenciômetro - 4 unidades

Protoboard - 1 unidade

Estes componentes serão descritos a seguir.

#### **3.1.1 Garra**

Ela é constituída em duas partes: a primeira é o esqueleto que constitui-se de peças de madeira do tipo MDF (Medium Density Fiberboard), um termo oriundo do inglês que traduzido à nossa língua significa fibra de média densidade, cortadas a laser, sendo que as mesmas possuem dimensões em escalas centimétricas possuindo uma espessura de 3 mm; a segunda parte é o uso do servo motor que tem a possibilidade de fazer o controle dos movimentos e o Arduino que atua como o mestre, porque ele é quem gerencia as informações.

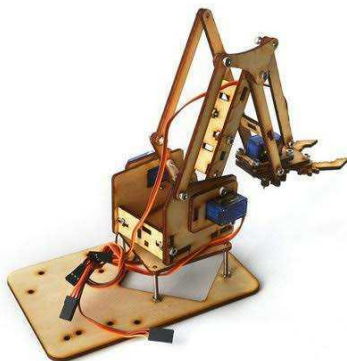


Figura 1.1 Foto de garra com servo embutido

### 3.1.2 Servo Motor

Neste projeto foi utilizado um servo motor para Arduino, cujo o modelo do mesmo é Micro Servo 9G SG90, considerado um motor fraco se comparado a outros de mesmo uso. Seu princípio de funcionamento rege ao movimento angular, seu movimento vai de 0° a 180°, esse controle de ângulo, pelo Arduino, é feito através das portas PWM (Pulse With Modulation) ou analógicas, formalmente é utilizado um comando chamado map, que na linguagem Arduino significa Regra de três, para converter dados. Este tipo de comando é necessário quando se deseja converter uma informação analógica em outra com valores diferentes, porém, mantendo a proporcionalidade entre os mesmos e é frequentemente utilizado nos trabalhos relacionados a controle.



Figura 1.2 Foto de um servo motor

### 3.1.3 Arduino

Neste projeto utilizou-se um Arduino do modelo Uno V3.0. Essa placa possui um microcontrolador, cujo o mesmo utilizado é o modelo ATmega328, atuando como um computador em escala reduzida, já que o mesmo possuindo internamente: um microprocessador com uma velocidade de clock de 16MHz, uma memória do tipo SRAM(Static Random Access Memory, em português significa memória de acesso randômico estático) de 2KB, uma memória do tipo EEPROM(Electrically-Erasable Programmable Read-Only Memory, traduzindo fica memória de somente leitura programável apagável eletricamente) de 1KB e uma memória do tipo Flash(Semelhante à EEPROM) de 32KB. Ele contém 14 portas digitais (6 podem ser utilizadas como PWM) e é um produto open Source (software aberto), implicando que qualquer um pode fazer o seu modelo de Arduino e adapta-lo. Sua criação data do ano de 2009 e seu local de criação foi na Itália. Com ele você pode fazer o projeto que você quiser



Figura 1.3 Foto de um Arduino

### 3.1.4 Potenciômetro

Se trata de um resistor que pode ter seu valor alterado através de uma alavanca nele presente possuindo o intuito de controlar a corrente que se passa para o circuito e contendo diversas aplicações. Estas peças se dividem em vários modelos e em dois tipos.

Os modelos são classificados com valores de alcance e valores de tensões. Dentre os mais comuns de alcance temos modelos de: 0 a 100 $\Omega$ ; 0 a 500 $\Omega$ ; 0 a 1k $\Omega$ ; 0 a 10K $\Omega$ , enquanto para tensão temos os modelos mais utilizados para o Arduino de 5 volts e 10 volts.

Os tipos de potenciômetro existentes são os lineares e os logaritmos. Os lineares são os potenciômetros cuja variação ocorre de forma proporcional ao movimento da alavanca, isso fica mais detalhado na figura a seguir

Os potenciômetros do tipo logaritmo funcionam da seguinte maneira: quando se movimenta a alavanca o valor da resistência varia obedecendo uma equação logarítmica, temos como exemplo a figura:

Vale ressaltar que, diferente dos outros componentes utilizados, pode ser utilizado qualquer modelo e tipo nesta ocasião.



Figura 1.4 Foto de um potenciômetro

Como pode ver o potenciômetro possui três pernas, sendo que quando elas estão para cima o esquema fica exposto da seguinte forma: o esquerdo com polaridade positiva, o do meio o sinal e o do lado direito contendo polaridade negativa.

### 3.1.5 Protoboard

Protoboard existem para cumprir uma função: Aumentar a quantidade de pontos

de acesso, assim podendo conectar mais coisas em lugar ao mesmo tempo. Uma outra que não se pode deixar para trás é o fato de a mesma poder ser utilizada para a montagem dos circuitos eletrônicos deixando-os com uma aparência mais organizada. No nosso projeto foi usada uma parte do modelo de 830 pontos. Esta parte possui dois canais de encaixe cujo objetivo era alimentar a fase e o neutro dos servos.

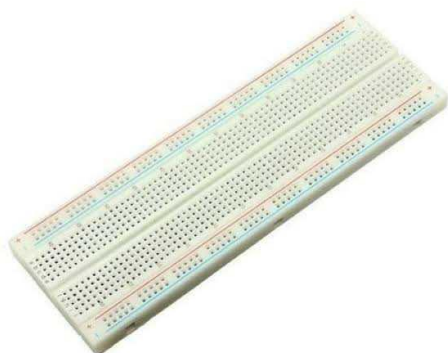


Figura 1.5 Foto de uma protoboard de 830 pontos

## **3.2 Construção e funcionamento**

Para a execução do projeto foi definido em duas etapas: uma baseada na montagem e conexão e outra com base na programação, que serão descritas a seguir:

### **3.2.1 Montagem e Conexão**

Contamos com uma garra, cujo seu esqueleto já estava finalizado. Precisou-se apenas colocar os Servos nos locais indicados e aperta-los ou aparafusa-los. Em seguida conectou-se os pinos de Sinal (geralmente Laranja) nas portas PWM do Arduino ficando a seguinte ordem: Servo da pinça conectado à porta 11, Servo direito na porta 10, Servo da base conectado à porta 9 e Servo esquerdo conectado à porta 6. Enquanto isso os pinos de fase (Vermelho) e neutro (Marrom) foram conectadas em uma protoboard a parte, pois não há saídas suficientes na placa e também que projetos do tipo garra utilizando servos costuma-se utilizar uma fonte de alimentação externa ao Arduino cujo a mesma deva possuir 5 volts e uma corrente igual ou superior a 2 amperes. Já a placa de potenciômetros foi diferente sua conexão, por se tratar de possuir três pernas (fase, sinal e neutro) deveríamos conectar 12 fios, 4 no Arduino e 8 na protoboard sendo 4 fios fase e 4 fios neutro, isso resultaria num gasto desnecessário em conexão, então foi optado por soldar as pernas, com isso poderia ser utilizado somente um único fio para fase e um fio para o neutro resultando em uma placa contendo 6 fios tendo, agora, 4 fios de sinais(conectada ao Arduino), 1 de fase e 1 de neutro(sendo estes dois conectados à protoboard). As portas de sinais dos potenciômetros foram conectadas nas portas Analógicas A0, A1, A2, A3.

### **3.2.2 Programação**

Toda a programação do sistema foi desenvolvida no IDE do Arduino versão

1.8.10. Para podermos prosseguir no projeto antes devemos pensar na lógica a ser utilizada que é de extrema importância em qualquer programa que você for criar na sua vida. Neste caso utilizaremos a seguinte lógica o potenciômetro irá enviar um valor x que deve ser reproduzido na garra de forma proporcional ao alcance do mesmo valor, ou seja, se movermos o potenciômetro em 1/3 de seu range o servo deverá locomover-se em 1/3 de sua angulação máxima ou, nesse caso, ele se locomoverá 60°. Todavia Nem todos os servos terão 180° de alcance, porque algum deles, no próprio sistema, terá limitação de seu movimento individual, tais como: pinça se abrir até um certo ponto, ela se abaixar até tocar o chão, entre outros.

A programação está exposta abaixo, comentada.

```
1 #include <Servo.h> // Declaramos a biblioteca a ser utilizada, como nesse caso é a servo, o Arduino já sabe que estamos, obviamente, trabalhando com um
2 Servo s1, s2, s3, s4; // Nesta parte se cria o(s) objeto(s) servo(s)
3 int p1 = A0, p2 = A1, p3 = A2, p4 = A3; // Nesta parte se declara as variáveis analógicas que serão utilizadas como portas de leitura
4 int v1, v2, v3, v4; // Variáveis a ser utilizada na escrita dos ângulos dos servos
5 void setup() {
6   Serial.begin(9600); // Utilizamos a comunicação Serial para analisar os valores e corrigir o alcance dos movimentos para evitar problemas futuros como perda da autonomia
7   s1.attach(11); // Com o comando attach podemos ancorar um objeto criado na programação a uma porta tanto Analógica ou Digital (nesse caso deve-se utilizar as portas pwm(11, 10, 9, 8, 5, 3))
8   s2.attach(10); // Aqui acontece o mesmo que na linha anterior
9   s3.attach(9);
10  s4.attach(6);
11 }
12 void loop() {
13   v1 = analogRead(p1); // Igualamos uma das variáveis do tipo inteira ao valor recebido pela porta Analógica
14   v1 = map(v1, 0, 1023, 55, 180); // Utilizando o comando map, que funciona como uma regra de três, podemos transformar um valor em outro
15   s1.write(v1); // servo pinça, B^ // Aqui pedimos ao sistema que a informação seja escrita na porta conectada à algum servo
16   delay(15); // Utilizamos o delay para que seja efetuado movimentos mais harmônicos, de uma cotidiana de dizer, ele se move com mais fluidez
17   v2 = analogRead(p2); // O que acontece a partir daqui, até a linha 29, foi explicado nas últimas quatro linhas
18   v2 = map(v2, 0, 1023, 0, 60);
19   s2.write(v2); // servo C^
20   delay(15);
21   v3 = analogRead(p3);
22   v3 = map(v3, 0, 1023, 180, 0);
23   s3.write(v3); // servo base, D^
24   delay(15);
25   v4 = analogRead(p4);
26   v4 = map(v4, 0, 1023, 90, 180);
27   s4.write(v4); // servo A^
28   delay(15);
29   //Serial.print("Valor:"); // Aqui enviamos um caractere a ser escrito no monitor Serial
30   //Serial.println(v1); // Aqui o valor, no momento atual da leitura, da variável será escrita no monitor Serial
31   //delay(200); // Como a resposta do arduino é muito rápida, diminuimos a velocidade de reprodução das informações a serem lidas
32 }
```

Figura 1.6 Foto da programação compilada e compartilhada

## 4. CONSIDERAÇÕES FINAIS

Este é somente uma das muitas maneiras de controlar uma garra, com motor de movimento sendo o servo motor. Pela situação descrita acima os problemas mais decorrentes de acontecer não costuma ocorrer na parte de software e sim na de montagem, entre os mais comuns, pinagem errada e fonte de alimentação insuficiente ou estragada, nesse caso recomenda-se carregadores de celular, por ser mais prático, que possuem uma corrente acima de 2A e que seja de cabo usb móvel, pois você não deseja estragar o carregador de seu celular, sempre use um reserva ou compre, se necessário.

Todas as fotos, com exceção da programação, foram tiradas do Google imagens

## 5. REFERÊNCIAS

<https://www.techtudo.com.br>