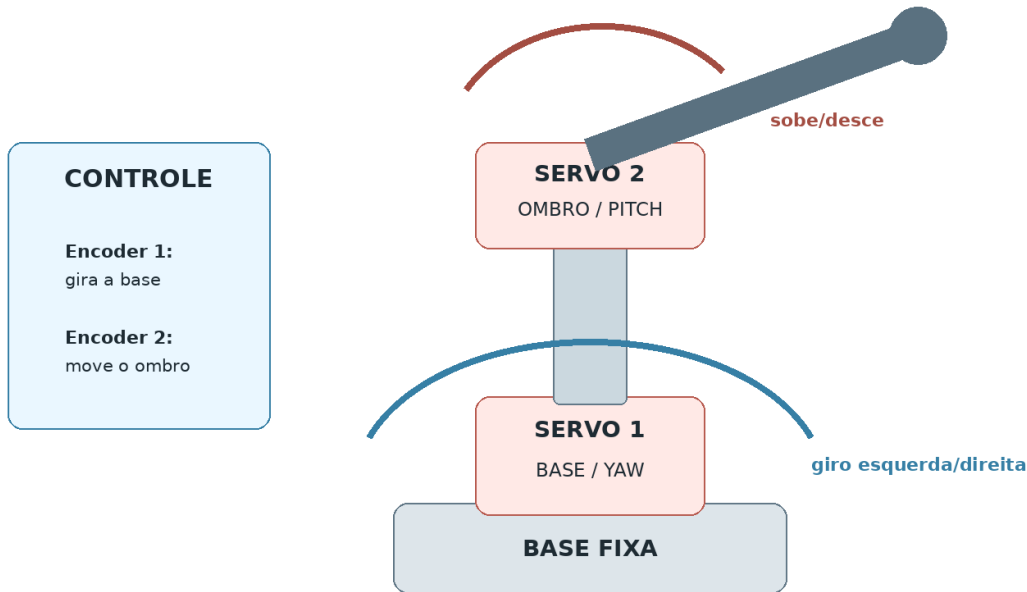


BRAÇO ROBÓTICO CONTROLADO POR ENCODER ROTATIVO

Arduino Uno | Servos de 180° e 360° | Alimentação por pilhas

Modelo 2 - braço de 2 graus de liberdade



Observação: 2 servos = 2 eixos. Para posicionamento 3D mais completo, adicione um 3º servo no cotovelo.

Dois projetos no mesmo manual

Modelo 1: um servo de 180° + um encoder + chave liga/desliga.

Modelo 2: dois servos + dois encoders para dois eixos de movimento. Inclui adaptação opcional para servo contínuo de 360°.

Nível: iniciante a intermediário | foco em montagem, programação, segurança e resolução de problemas

1. Antes de construir: três ideias essenciais

1) Servo de 180° = controle de posição

É o tipo mais indicado para articulações de um braço. Você manda um ângulo (por exemplo 30°, 90° ou 150°) e o servo tenta chegar e permanecer nessa posição.

2) Servo de 360° = controle de velocidade e sentido

Um servo contínuo não sabe “ir para 120°”. Normalmente 90 significa parar, valores menores giram em um sentido e maiores no outro. Para articulações com posição conhecida, prefira o servo de 180°.

3) Dois servos não dão posicionamento 3D completo

Com dois servos podemos construir dois graus de liberdade: por exemplo, girar a base para esquerda/direita e levantar/abaixar o braço. Para alcançar livremente pontos no espaço 3D, o passo seguinte é adicionar um terceiro servo no cotovelo.

O que o estudante aprenderá

- identificar Arduino Uno, servo motor, encoder rotativo, fonte e conexões de alimentação;
- montar um sistema alimentado por pilhas sem sobrecarregar o Arduino;
- usar um encoder como dispositivo de comando;
- programar posição de servo com a biblioteca Servo;
- instalar e usar a biblioteca Encoder;
- calibrar limites mecânicos e corrigir sentido de rotação;
- combinar dois eixos de movimento em um braço robótico simples.

2. Componentes e ferramentas

Modelo 1 - entrada

Item	Quantidade	Observação
Arduino Uno R3	1	Controlador principal
Servo 180° (ex.: SG90)	1	Para braço leve
Encoder rotativo incremental (ex.: KY-040)	1	Controla o ângulo
Chave ou botão travante liga/desliga	1	Deve manter a posição ON/OFF
Suporte para 6 pilhas AA	1	NiMH recarregáveis são recomendáveis
Conversor buck DC-DC 5 V / >=2 A	1	Alimenta o servo
Jumpers	conforme necessário	Macho-macho e/ou macho-fêmea
Protoboard	1	Opcional, útil para organização
Material estrutural	-	MDF, acrílico, impressão 3D, papelão rígido ou madeira leve

Modelo 2 - mais robusto

Item	Quantidade	Observação
Arduino Uno R3	1	Controlador
Servo 180°	2	Base + ombro
Encoder rotativo	2	Um por eixo
Buck 5-6 V	1	2-3 A para microservos; 4-5 A para servos de alto torque
Capacitor eletrolítico 470-1000 uF	1	Perto dos servos ajuda contra quedas de tensão
6 pilhas AA + suporte	1 conjunto	Preferir NiMH de boa capacidade
Chave liga/desliga travante	1	Corta a alimentação do conjunto
Estrutura reforçada	1	Impressão 3D, MDF ou acrílico

Sobre o “botão liga/desliga”

Um botão momentâneo comum só fica ativo enquanto está pressionado. Para economizar bateria sem circuitos extras, use uma chave deslizante, rocker ou botão travante (latching).

3. Como o sistema pensa

MÃO DO USUÁRIO -> ENCODER -> ARDUINO -> SERVO -> MOVIMENTO DO BRAÇO

Encoder rotativo

O encoder incremental produz pulsos quando giramos o botão. O Arduino conta esses pulsos para decidir se deve aumentar ou diminuir o ângulo do servo.

Pino KY-040	Função	Modelo 1
CLK	pulso A	D2
DT	pulso B	D3
SW	botão interno do encoder	opcional
+	alimentação	5V
GND	referência	GND

Servo motor

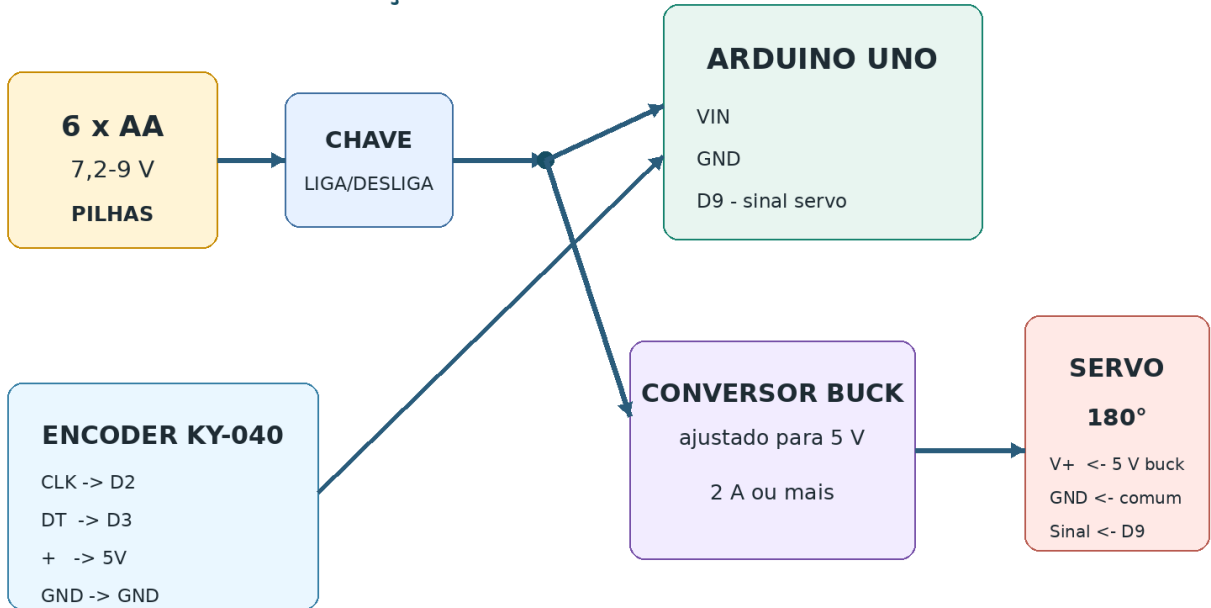
Fio típico	Função	Ligação
Vermelho	V+	5-6 V da fonte/buck
Marrom ou preto	GND	GND comum
Laranja/amarelo/branco	Sinal	pino digital do Arduino

Por que o servo não deve ser alimentado pelo pino 5V do Arduino?

Servos podem pedir picos de corrente muito maiores do que o regulador do Arduino consegue fornecer. Isso pode reiniciar a placa, gerar movimentos erráticos ou danificar componentes. O Arduino fornece o SINAL; a fonte externa fornece a POTÊNCIA.

4. Alimentação por pilhas e segurança

Modelo 1 - alimentação e conexões



Regra de ouro: o GND do Arduino, do buck, do servo e do encoder deve ser comum.

Arquitetura recomendada

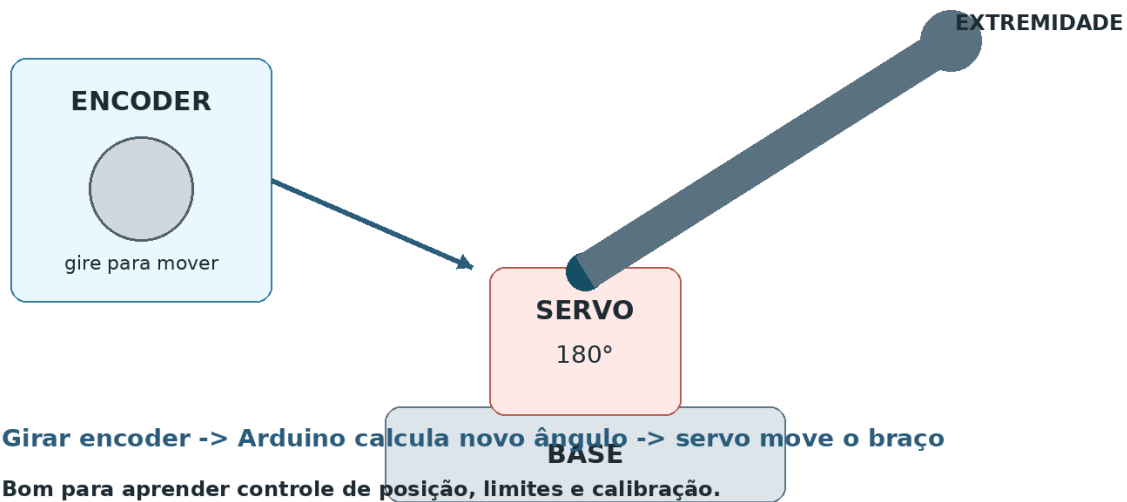
1. Coloque a chave liga/desliga logo após o suporte de pilhas.
2. Depois da chave, divida a alimentação em dois caminhos.
3. Caminho A: leve a tensão das 6 pilhas ao VIN do Arduino Uno.
4. Caminho B: leve a mesma bateria ao conversor buck e ajuste a saída para 5,0 V antes de conectar o servo.
5. Una todos os GNDs: Arduino, buck, servo e encoder.
6. Só ligue o sistema depois de revisar polaridade e tensão com multímetro, quando disponível.

Nunca faça

Não conecte 9 V diretamente ao servo. Não inverta + e GND. Não prenda o braço mecanicamente enquanto o servo tenta se mover. Não deixe fios desencapados encostarem entre si.

5. Modelo 1 - construção mecânica de um eixo

Modelo 1 - braço de 1 eixo



Montagem sugerida

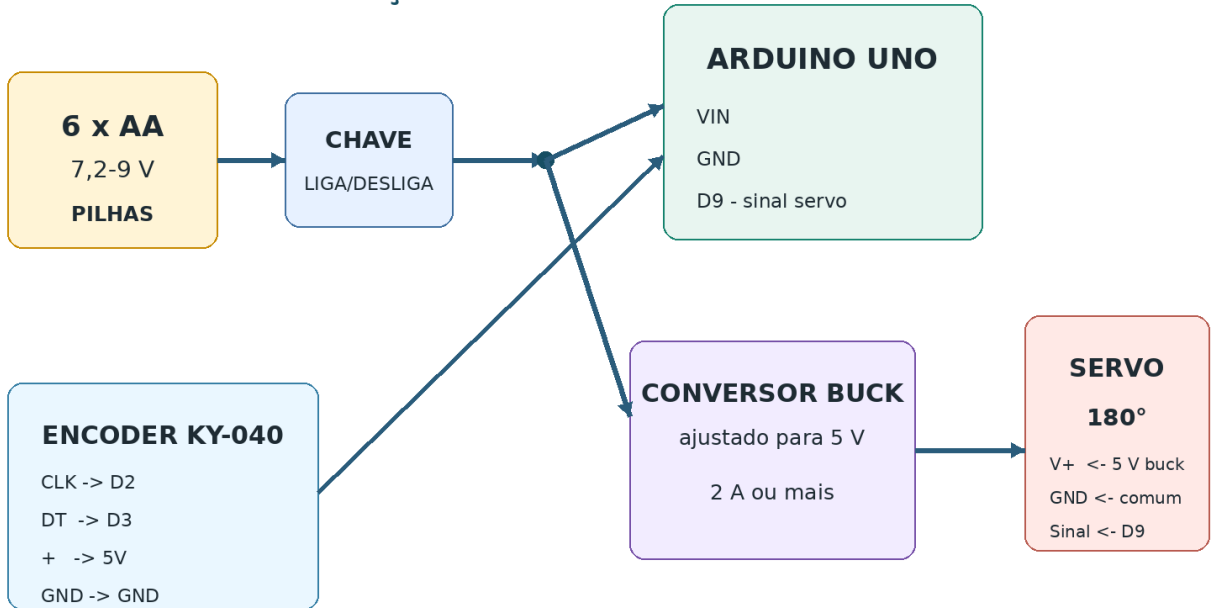
1. Faça uma base firme com MDF, acrílico, impressão 3D ou papelão rígido reforçado.
2. Fixe o servo na base de forma que seu eixo fique livre.
3. Com o servo ainda desligado, encaixe provisoriamente o horn (braço plástico do servo).
4. Construa um braço leve de 15 a 25 cm. Para SG90, evite peças pesadas.
5. Fixe o braço ao horn somente depois da calibração inicial em 90°.
6. Monte o encoder em uma pequena caixa de controle para o estudante girar confortavelmente.
7. Deixe a chave liga/desliga acessível.

Por que calibrar em 90°?

Se montarmos o braço mecanicamente em uma posição conhecida quando o servo está no centro, fica mais fácil evitar que o braço force contra a base ao chegar nos limites.

6. Modelo 1 - ligações elétricas

Modelo 1 - alimentação e conexões



Regra de ouro: o GND do Arduino, do buck, do servo e do encoder deve ser comum.

Tabela de conexões

Componente	Pino	Conectar em
Servo 1	Sinal	D9
Servo 1	V+	5 V do buck
Servo 1	GND	GND comum
Encoder	CLK	D2
Encoder	DT	D3
Encoder	+	5V do Arduino
Encoder	GND	GND do Arduino
Bateria	+	chave -> VIN e buck
Bateria	-	GND comum

1. Mantenha a chave desligada.
2. Faça primeiro GND e alimentação.
3. Depois conecte CLK e DT do encoder.
4. Conecte somente o fio de sinal do servo ao D9.
5. Confirme que V+ do servo vem do buck, não do 5V do Arduino.
6. Revise tudo antes de ligar.

7. Preparando o Arduino IDE

1. Conecte o Arduino Uno ao computador por um cabo USB de dados.
2. Abra o Arduino IDE.
3. Selecione Ferramentas -> Placa -> Arduino Uno.
4. Selecione a porta COM correspondente.
5. A biblioteca Servo normalmente já acompanha o ambiente Arduino.
6. Abra o Gerenciador de Bibliotecas, procure “Encoder” e instale a biblioteca Encoder (Paul Stoffregen).

Primeiro teste: centralizar o servo

```
#include <Servo.h>

Servo serv01;

void setup() {
  serv01.attach(9);
  serv01.write(90);
}

void loop() {
}
```

Faça isto antes de prender o braço ao horn

Carregue o código, deixe o servo ir para 90°, desligue a alimentação e só então fixe mecanicamente o braço na posição central desejada.

8. Código completo - 1 servo de 180° + 1 encoder

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

Servo servoBase;
Encoder encoder(2, 3);

const int PINO_SERVO = 9;
int angulo = 90;
long ultimaPosicao = 0;

void setup() {
  Serial.begin(9600);
  servoBase.attach(PINO_SERVO);
  servoBase.write(angulo);

  encoder.write(0);
  Serial.println("Braco pronto. Gire o encoder.");
}

void loop() {
  long posicao = encoder.read() / 4;

  if (posicao != ultimaPosicao) {
    long diferenca = posicao - ultimaPosicao;

    angulo = angulo + (diferenca * 2);
    angulo = constrain(angulo, 10, 170);

    servoBase.write(angulo);

    Serial.print("Angulo: ");
    Serial.println(angulo);

    ultimaPosicao = posicao;
  }
}
```

O que modificar

- Se o encoder girar ao contrário, troque os pinos 2 e 3 na linha `Encoder encoder(2, 3)`.
- Se o braço mover rápido demais, troque “* 2” por “* 1”.
- Se o braço puder bater na estrutura, reduza os limites 10 e 170, por exemplo para 25 e 150.
- Se cada “clique” do seu encoder gerar quantidade diferente de pulsos, talvez seja necessário alterar o divisor /4.

9. Teste, calibração e adaptação para servo 360°

Checklist do modelo 1

- Ao ligar, o servo vai para a posição central sem bater na estrutura.
- Girar o encoder em um sentido aumenta o ângulo.
- Girar no outro sentido diminui o ângulo.
- O servo não treme continuamente quando parado.
- O Arduino não reinicia quando o servo começa a mover.
- A chave desliga completamente o conjunto.

Se usar servo contínuo de 360°

Troque a ideia de “ângulo” pela ideia de “velocidade”. O encoder pode aumentar ou diminuir um comando de -90 a +90. O valor 0 representa parada; valores positivos giram em um sentido e negativos no outro.

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

Servo servo360;
Encoder encoder(2, 3);
int comando = 0;
long ultima = 0;

void setup() {
  servo360.attach(9);
  servo360.write(90); // parar
}

void loop() {
  long p = encoder.read() / 4;

  if (p != ultima) {
    comando += (p - ultima) * 5;
    comando = constrain(comando, -90, 90);

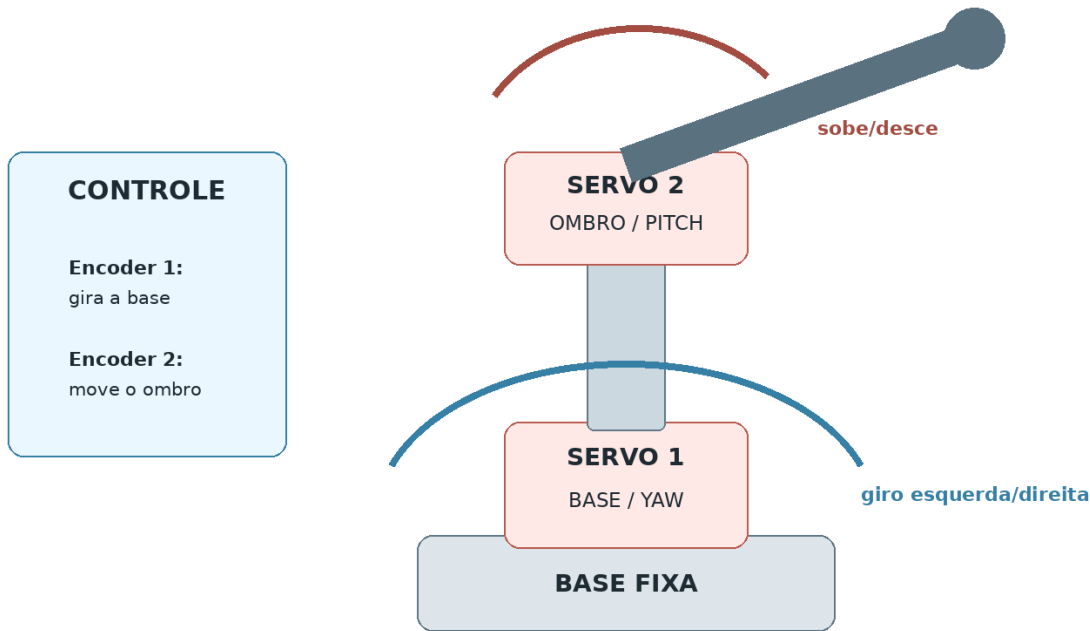
    servo360.write(90 + comando);
    ultima = p;
  }
}
```

Limitação importante

Servo contínuo de 360° não fornece posição angular conhecida. Para uma articulação de braço que precisa ir exatamente para um ponto, use servo posicional de 180° ou um motor com encoder de posição.

10. Modelo 2 - braço mais robusto com dois eixos

Modelo 2 - braço de 2 graus de liberdade



Observação: 2 servos = 2 eixos. Para posicionamento 3D mais completo, adicione um 3º servo no cotovelo.

O segundo modelo usa dois servos posicionais. O servo da base controla o giro esquerda/direita (yaw). O servo do ombro controla subida/descida (pitch). Cada eixo recebe seu próprio encoder.

Nome correto do movimento

Este modelo tem 2 graus de liberdade (2 DOF). Ele já permite um movimento espacial muito mais interessante, mas não substitui um braço de 3 eixos. Para 3D mais completo, adicione um servo de cotovelo.

Escolha de servos

- Dois SG90: bom para maquete leve, custo baixo, torque limitado.
- MG90S: engrenagens metálicas, um pouco mais robusto.
- MG996R ou similar: maior torque, exige fonte 5-6 V de alta corrente e estrutura resistente.

11. Modelo 2 - montagem mecânica

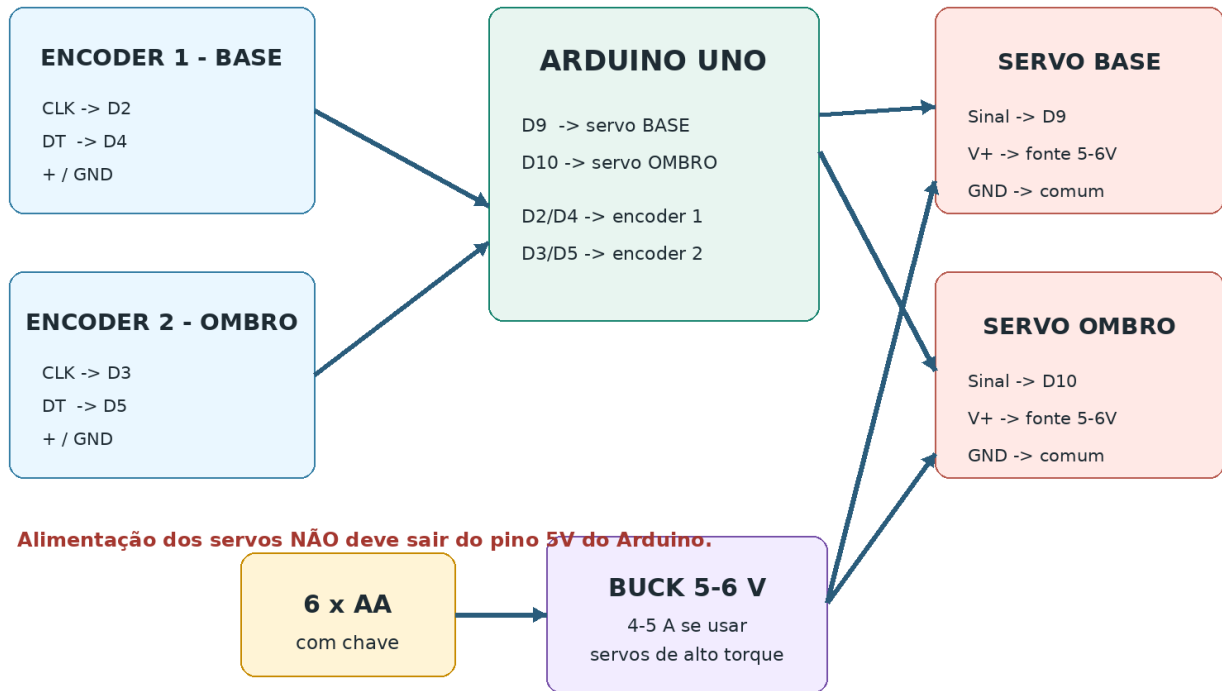
1. Construa uma base larga e pesada o suficiente para não tombar.
2. Instale o servo da base com seu eixo vertical. Ele será responsável pelo giro esquerda/direita.
3. Fixe uma torre ou suporte sobre o horn do servo da base.
4. Na parte superior da torre, instale o segundo servo com eixo horizontal.
5. Prenda o braço principal ao horn do servo do ombro.
6. Antes de apertar parafusos, envie 90° para os dois servos e alinhe a estrutura em uma posição neutra.
7. Verifique manualmente se nenhum cabo fica esticado quando a base gira.
8. Monte os dois encoders na caixa de controle e identifique-os como BASE e OMBRO.

Estrutura

Se o Espaço Cria usar a impressora 3D, vale projetar suportes de servo com furos para parafusos M2/M3 e um braço modular. Para primeiro protótipo, MDF fino ou papelão rígido reforçado funciona melhor do que peças pesadas.

12. Modelo 2 - ligações elétricas

Modelo 2 - dois eixos, dois servos e dois encoders



Componente	Sinal	Pino Arduino
Servo base	sinal	D9
Servo ombro	sinal	D10
Encoder 1	CLK	D2
Encoder 1	DT	D4
Encoder 2	CLK	D3
Encoder 2	DT	D5

Alimentação robusta

Para dois servos de alto torque, use buck/fonte 5-6 V com capacidade de aproximadamente 4-5 A e um capacitor de 470 a 1000 uF próximo aos servos. Para dois micros servos leves, 2-3 A costuma ser suficiente.

GND comum é obrigatório

Mesmo usando uma fonte separada para os servos, o GND dessa fonte precisa estar conectado ao GND do Arduino. Sem referência comum, o sinal de controle pode ficar instável.

13. Código completo - dois servos e dois encoders

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

Servo servoBase;
Servo servoOmbro;

Encoder encBase(2, 4);
Encoder encOmbro(3, 5);

int angBase = 90;
int angOmbro = 90;

long posBaseAnt = 0;
long posOmbroAnt = 0;

void setup() {
  Serial.begin(9600);

  servoBase.attach(9);
  servoOmbro.attach(10);

  servoBase.write(angBase);
  servoOmbro.write(angOmbro);

  encBase.write(0);
  encOmbro.write(0);
}

void loop() {
  long posBase = encBase.read() / 4;
  long posOmbro = encOmbro.read() / 4;

  if (posBase != posBaseAnt) {
    angBase += (posBase - posBaseAnt) * 2;
    angBase = constrain(angBase, 10, 170);
    servoBase.write(angBase);
    posBaseAnt = posBase;
  }

  if (posOmbro != posOmbroAnt) {
    angOmbro += (posOmbro - posOmbroAnt) * 2;
    angOmbro = constrain(angOmbro, 20, 150);
    servoOmbro.write(angOmbro);
    posOmbroAnt = posOmbro;
  }
}
```

```
Serial.print("Base: ");  
Serial.print(angBase);  
Serial.print(" | Ombro: ");  
Serial.println(angOmbro);  
}
```

Não copie os limites mecanicamente

10-170° e 20-150° são valores iniciais. O limite correto depende da estrutura construída. Comece com faixa estreita e aumente aos poucos.

14. Diagnóstico de problemas

Sintoma	Causa provável	O que verificar
Servo treme	fonte fraca / ruído	fonte 5-6 V, capacitor, GND comum
Arduino reinicia	servo puxando corrente do Arduino	alimente servo pelo buck
Encoder pula valores	bounce / montagem / divisor	teste /4, conexões e biblioteca
Encoder gira invertido	ordem A/B trocada	troque CLK e DT no código
Servo bate na estrutura	limites grandes	reduza constrain()
Servo não move	sinal/potência incorretos	D9/D10, GND e 5-6 V
360° não para em 90	servo precisa de calibração	ajuste 88-92 até encontrar o neutro
Movimento lento sob carga	torque insuficiente	reduza braço/peso ou use servo mais forte

Procedimento de depuração

1. Desligue o braço e retire a carga mecânica se houver esforço.
2. Teste cada servo sozinho com o código de centralização.
3. Teste cada encoder sozinho imprimindo seu valor no Monitor Serial.
4. Meça a saída do buck: deve permanecer próxima de 5-6 V mesmo quando o servo se move.
5. Monte novamente um componente de cada vez.

15. Transformando o braço em projeto de aprendizagem baseada em problemas

Depois de dominar o controle manual, o desafio deixa de ser “mover um servo” e passa a ser “resolver um problema”. O estudante precisa escolher mecanismo, sensores e lógica.

Problema da comunidade/escola	Possível função do braço	Próximo componente
Manipular objeto quente ou contaminado	afastar a mão do usuário	garra + 3º servo
Organizar pequenas peças no laboratório	pegar e deslocar	sensor de presença + garra
Abrir/fechar ventilação de miniestufa	acionar mecanismo	DHT11 + controle automático
Apontar sensor/câmera para áreas diferentes	orientar plataforma	sensor de luz ou câmera
Acessibilidade para acionar objetos distantes	empurrar/puxar comando	botão adaptado / joystick
Monitorar colmeia com sensor móvel	reposicionar sensor sem abrir estrutura	DHT11/temperatura + LoRa

Desafio final

MISSÃO

Projete uma adaptação do braço para resolver um problema observado na escola ou comunidade. A equipe deve justificar: qual movimento é necessário, qual servo é adequado, como será alimentado, quais limites garantem segurança e que sensor poderia automatizar a tarefa.

Evolução para 3 eixos

- Servo 1: base - yaw (esquerda/direita).
- Servo 2: ombro - pitch (sobe/desce).
- Servo 3: cotovelo - pitch (aproxima/afasta).
- Servo 4 opcional: garra (abre/fecha).

16. Checklist de montagem e avaliação

Antes de ligar

- ☐ Polaridade das pilhas conferida.
- ☐ Chave liga/desliga instalada no positivo da bateria.
- ☐ Buck ajustado para 5-6 V.
- ☐ Servo não está ligado diretamente aos 9 V das pilhas.
- ☐ Todos os GNDs estão comuns.
- ☐ Braço consegue se mover manualmente sem travar a estrutura.
- ☐ Fios estão longe das partes móveis.

O estudante demonstra domínio quando consegue

- ☐ explicar a função do encoder, Arduino e servo;
- ☐ identificar corretamente V+, GND e sinal;
- ☐ alterar os limites do servo no código;
- ☐ corrigir o sentido de um encoder;
- ☐ explicar por que os servos usam alimentação externa;
- ☐ calibrar posição neutra e limites mecânicos;
- ☐ usar dois encoders para comandar dois eixos;
- ☐ propor uma aplicação útil para o braço robótico.

Próximo passo recomendado

Depois que o modelo de dois eixos estiver estável, adicione um terceiro servo no cotovelo e um quarto na garra. Em seguida, substitua um dos comandos manuais por um sensor e desafie os alunos a automatizar uma tarefa.

Referências de programação

Bibliotecas utilizadas: Servo (Arduino) e Encoder (Paul Stoffregen). Consulte os exemplos instalados na própria Arduino IDE e a documentação das bibliotecas para explorar recursos adicionais.