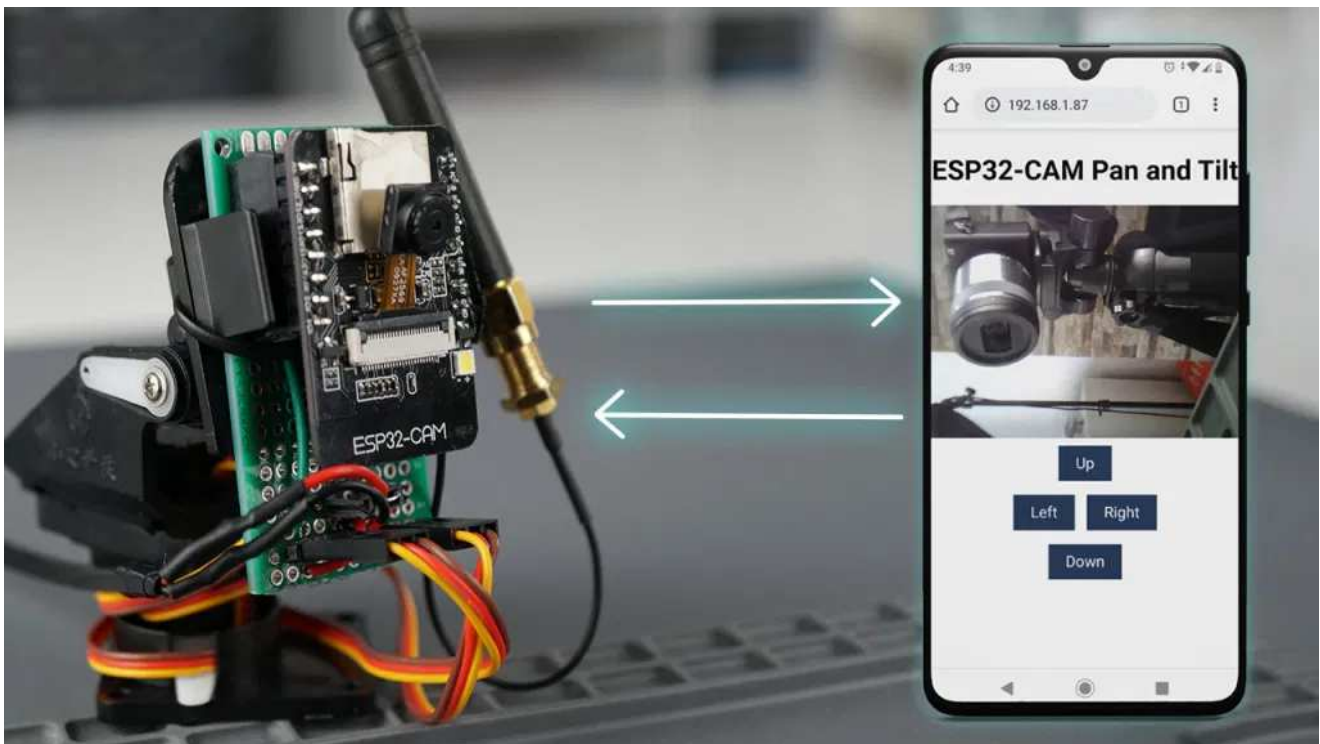


ESP32-CAM Pan e Tilt Video Streaming Web Server (2 Eixos)

Neste projeto, vamos anexar o ESP32-CAM a uma panela e suporte de inclinação com dois servomotores SG90. Com uma panela e uma câmera inclinada, você pode mover a câmera para cima, para baixo, para a esquerda e para a direita, isso é ótimo para vigilância. O ESP32-CAM hospeda um servidor web que mostra streaming de vídeo e botões para controlar os servo motores para mover a câmera.



Compatibilidade de placas: para este projeto, você precisa de uma placa de desenvolvimento de câmera ESP32 com acesso a dois GPIOs para controlar dois servomotores. Você pode usar: [ESP32-CAM AI-Thinker](#), [T-Journal](#) ou [TTGO T-Camera Plus](#).

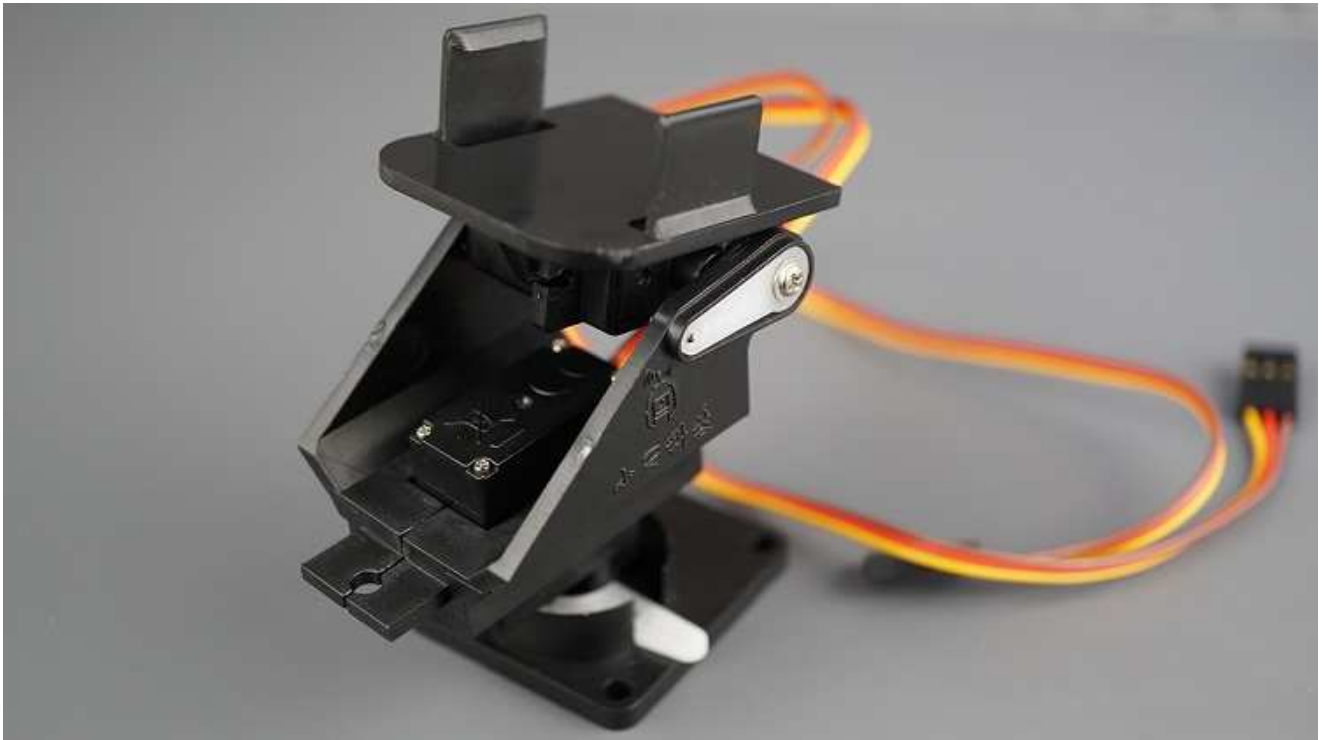
Peças necessárias

Para este projeto, usaremos as seguintes partes:

- [ESP32-CAM AI-Thinker com antena externa](#)
- [Suporte pan e inclinação com sg90 servomotores](#)
- [Placa de circuito de prototipagem \(opcional\)](#)
- [Fios de jumper](#)

Pan e Tilt Stand e Motores

Para este projeto, vamos usar uma panela e suporte de inclinação que já vem com dois servomotores SG90. O suporte é mostrado na figura a seguir.



Temos nossa posição de Banggood, mas você pode obter o seu de qualquer outra loja.

- [Pan e Tilt ficam com dois servomotores](#)

Alternativamente, você pode obter [dois servomotores SG90](#) e imprimir 3D seu próprio suporte.

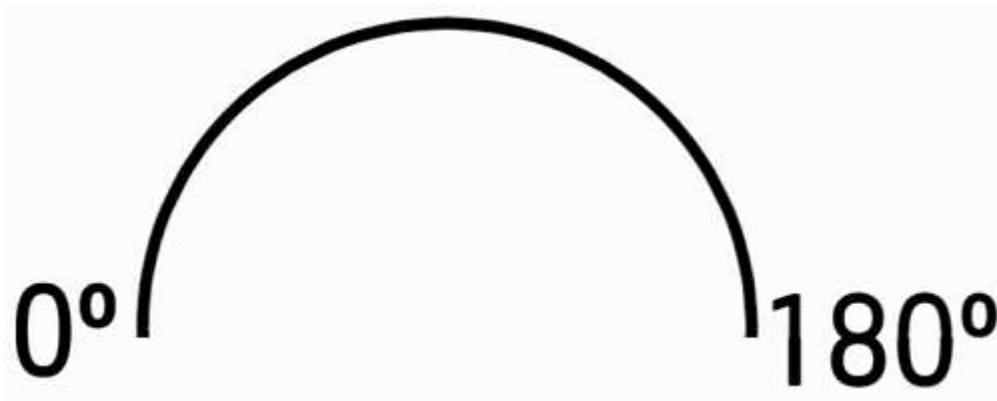
Servo motors têm três fios com cores diferentes:

Arame	Cor
Poder	Vermelho

GND	Preto ou marrom
Sinal	Amarelo, laranja ou branco

Como controlar um Servo?

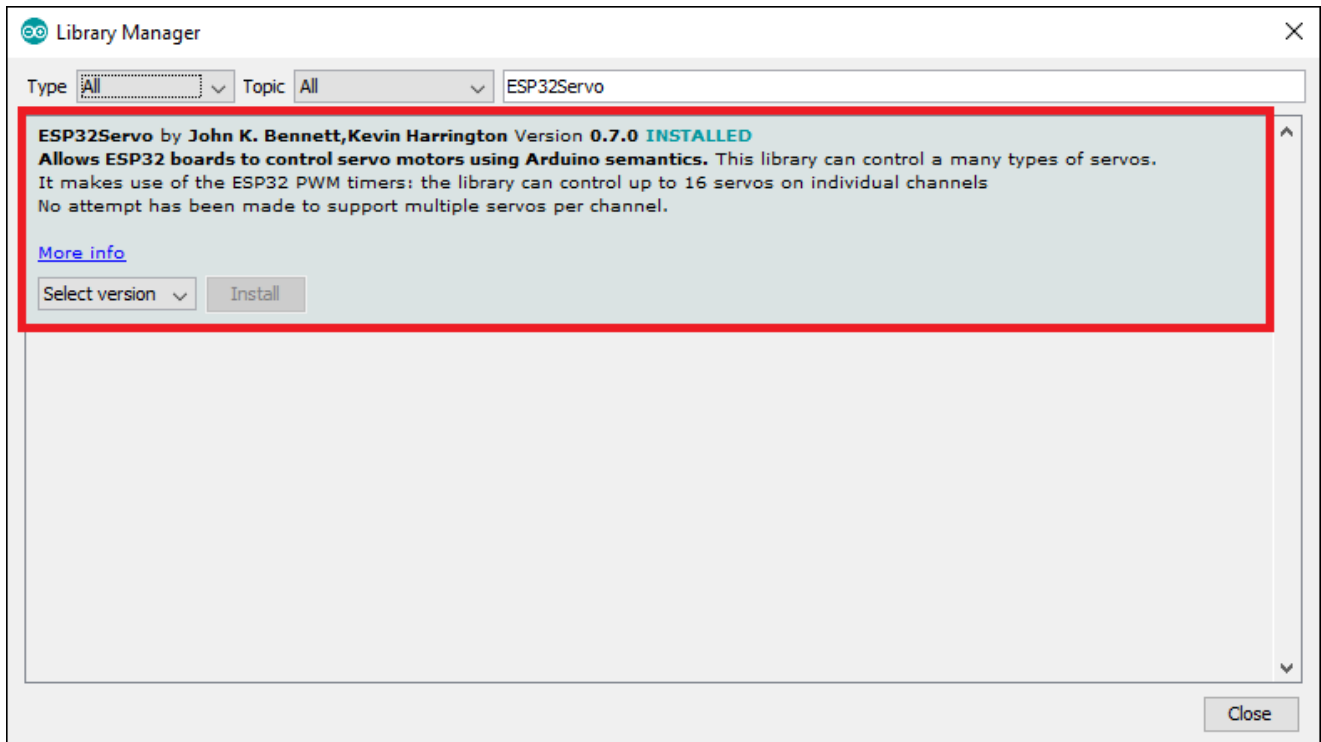
Você pode posicionar o eixo do servo em vários ângulos de 0 a 180°. Os servos são controlados usando um sinal de modulação de largura de pulso (PWM). Isso significa que o sinal PWM enviado ao motor determina a posição do eixo.



Para controlar o servo motor, você pode usar as capacidades PWM do ESP32 enviando um sinal com a largura de pulso apropriada. Ou você pode usar uma biblioteca para tornar o código mais simples. Usaremos a biblioteca **ESP32Servo**.

Instalando a Biblioteca ESP32Servo

Para controlar os servomotores, usaremos a biblioteca **ESP32Servo**. Certifique-se de instalar essa biblioteca antes de prosseguir. Em seu IDE Arduino, vá para **Sketch > Inclua Biblioteca > Gerenciar bibliotecas**. Procure **esp32Servo** e instale a biblioteca como mostrado abaixo.



Código

Copie o código a seguir para o seu IDE Arduino.

```

/*****
  Rui Santos
  Complete instructions at https://RandomNerdTutorials.com/esp3

  Permission is hereby granted, free of charge, to any person o
  The above copyright notice and this permission notice shall b
  *****/

```

```

#include "esp_camera.h"
#include <WiFi.h>
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h"           // disable brownout problems
#include "soc/rtc_cntl_reg.h" // disable brownout problems
#include "esp_http_server.h"
#include <ESP32Servo.h>

```

```
// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

#define PART_BOUNDARY "1234567890000000000000987654321"

#define CAMERA_MODEL_AI_THINKER
```

[Exibir código bruto](#)

Credenciais de rede

Insira suas credenciais de rede e o código deve funcionar imediatamente.

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

Como funciona o código

Vamos dar uma olhada nas partes relevantes para controlar os servo motores.

Defina os pinos aos quais os servomotores estão conectados. Neste caso, eles estão conectados aos GPIOs ESP32-CAM 14 e 15.

```
#define SERVO_1 14
#define SERVO_2 15
```

Criar Servo objetos para controlar cada motor:

```
Servo servoN1;
Servo servoN2;
Servo servo1;
Servo servo2;
```

Você deve estar se perguntando por que estamos criando quatro `Servo` objetos quando temos apenas dois servos. O que acontece é que a biblioteca `servo` que estamos usando atribui automaticamente um canal PWM a cada servo motor (`servoN1` → canal PWM 0; `servoN2` → canal PWM 1; `servo1` → canal PWM 2; `servo2` → canal PWM 3).

Os primeiros canais estão sendo usados pela câmera, então se mudarmos as propriedades desses canais PWM, teremos erros com a câmera. Então, nós vamos controlar `servo1` e `servo2` que usam os canais PWM 2 e 3 que não estão sendo usados pela câmera.

Defina a posição inicial dos servos.

```
int servo1Pos = 0;
int servo2Pos = 0;
```

Página web

O `INDEX_HTML` variável contém o texto HTML para construir a página da Web. As linhas a seguir exibem os botões.

```
<table>
  <tr><td colspan="3" align="center"><button class="button" onmou
  <tr><td align="center"><button class="button" onmousedown="toggl
  <tr><td colspan="3" align="center"><button class="button" onmou
</table>
```

Quando você clica nos botões, o `alternarCheckbox()` A função JavaScript é chamada. Ele faz uma solicitação em uma URL diferente, dependendo do botão clicado.

```
function toggleCheckbox(x) {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "/action?go=" + x, true);
```

```
xhr.send();  
}
```

Aqui estão as solicitações feitas dependendo do botão que está sendo pressionado:

Em cima:

```
/action?go=up
```

Abaixo:

```
/action?go=down
```

Esquerda:

```
/action?go=left
```

Direita:

```
/action?go=right
```

Atender solicitações

Então, precisamos lidar com o que acontece quando recebemos esses pedidos. Isso é o que é feito nas linhas a seguir.

```
if(!strcmp(variable, "up")) {  
    if(servo1Pos <= 170) {  
        servo1Pos += 10;  
        servo1.write(servo1Pos);  
    }  
}
```

```
Serial.println(servo1Pos);
Serial.println("Up");
}
else if(!strcmp(variable, "left")) {
    if(servo2Pos <= 170) {
        servo2Pos += 10;
        servo2.write(servo2Pos);
    }
    Serial.println(servo2Pos);
    Serial.println("Left");
}
else if(!strcmp(variable, "right")) {
    if(servo2Pos >= 10) {
        servo2Pos -= 10;
        servo2.write(servo2Pos);
    }
    Serial.println(servo2Pos);
    Serial.println("Right");
}
else if(!strcmp(variable, "down")) {
    if(servo1Pos >= 10) {
        servo1Pos -= 10;
        servo1.write(servo1Pos);
    }
    Serial.println(servo1Pos);
    Serial.println("Down");
}
```

Para mover um motor, chamar o `escrever()` função no `servo1` ou `servo2` objetos e passar o ângulo (0 a 180) como argumento. Por exemplo:

```
servo1.write(servo1Pos);
```

configuração()

No `configuração()`, defina as propriedades do servo motor: defina a frequência do sinal.


```
servo1.setPeriodHertz(50); // standard 50 hz servo  
servo2.setPeriodHertz(50); // standard 50 hz servo
```

Use o `anexar()` método para definir o servo GPIO e largura de pulso mínima e máxima em microsegundos.

```
servo1.attach(SERVO_1, 1000, 2000);  
servo2.attach(SERVO_2, 1000, 2000);
```

Coloque os motores na posição inicial quando o ESP32 inicializado.

```
servo1.write(servo1Pos);  
servo2.write(servo2Pos);
```

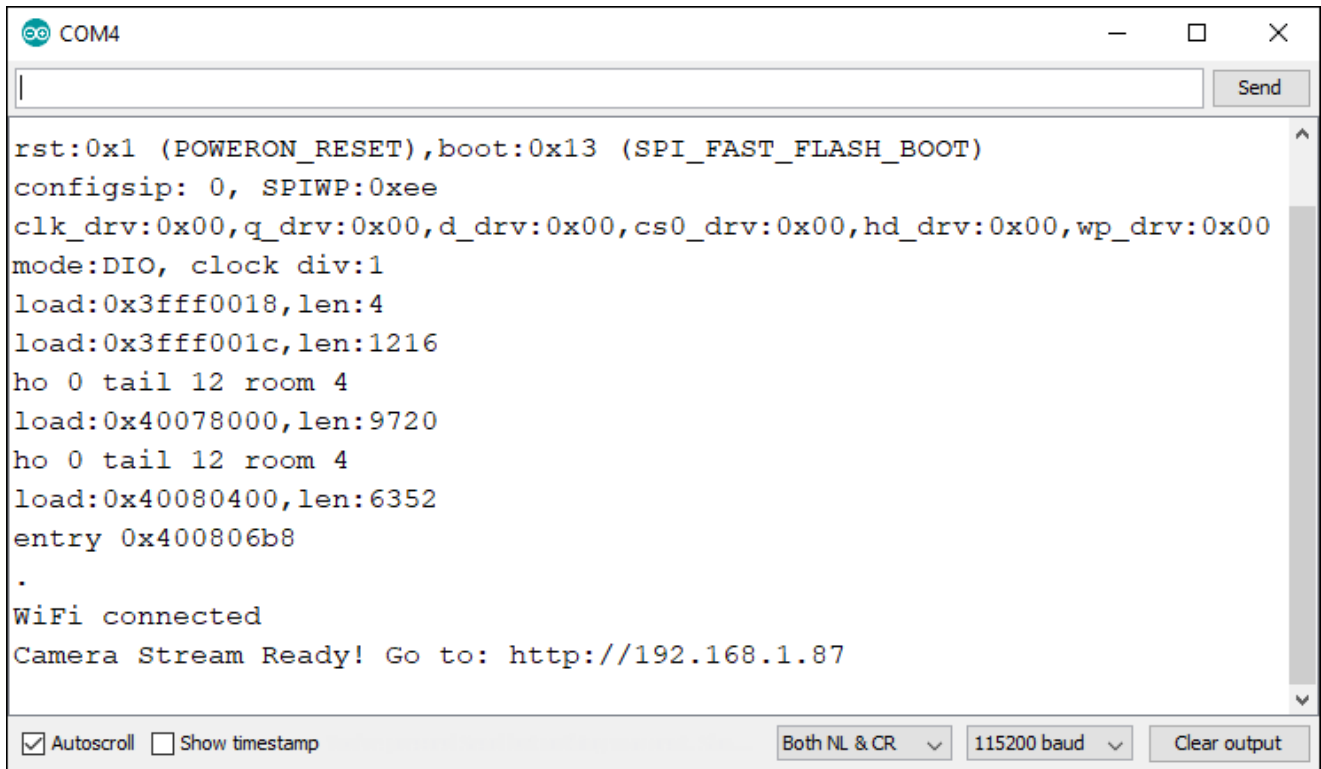
É assim que o código funciona quando se trata de controlar os servomotores.

Testando o Código

Depois de inserir suas credenciais de rede, você pode carregar o código para sua placa. Você pode usar um programador FTDI ou um programador ESP32-CAM MB. Leia um dos seguintes artigos:

- [Como Programar / Enviar Código para ESP32-CAM AI-Thinker \(Arduino IDE\) \(FTDI\)](#)
- [Código de upload para ESP32-CAM AI-Thinker usando o programador USB ESP32-CAM-MB \(maneira mais fácil\)](#)

Após o upload, abra o Serial Monitor para obter o endereço IP da placa.



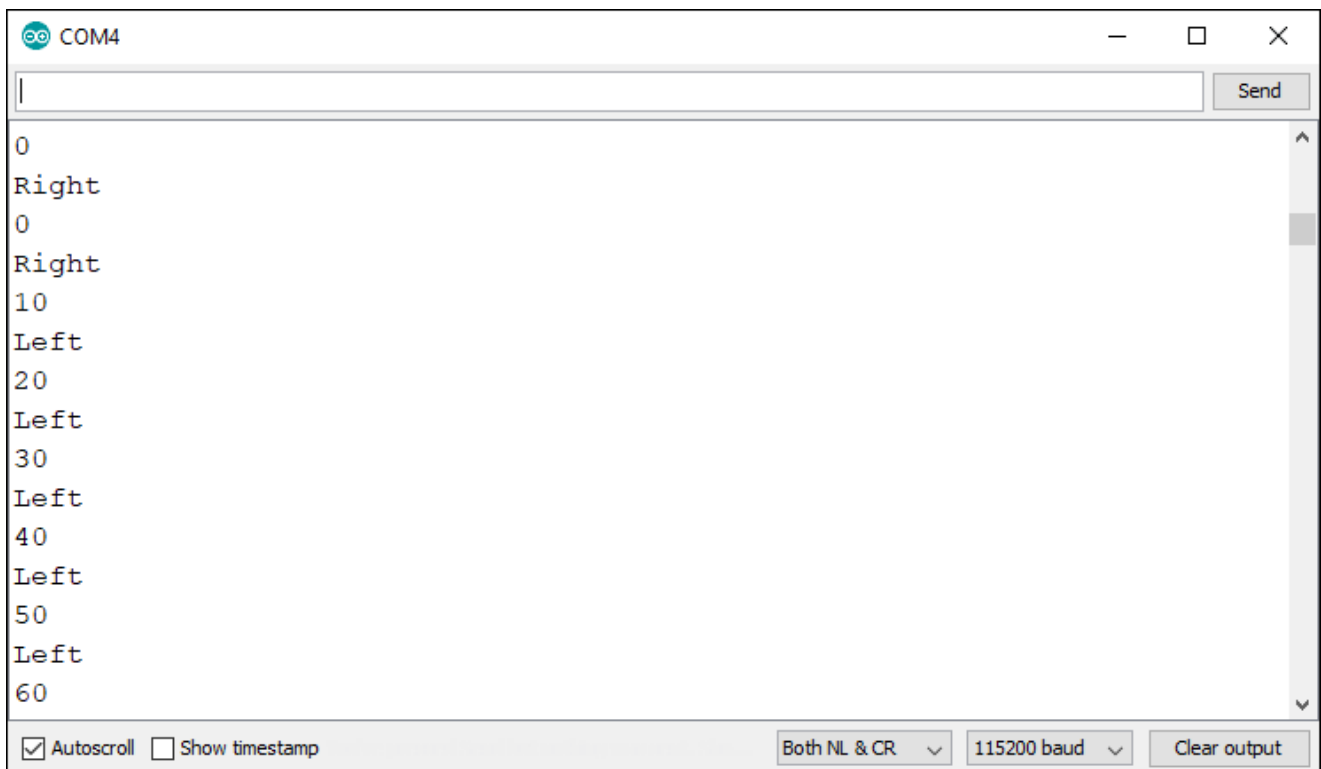
```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8
.
WiFi connected
Camera Stream Ready! Go to: http://192.168.1.87
```

COM4

☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

Nota: se estiver usando um programador FTDI, não se esqueça de desconectar o GPIO 0 do GND antes de abrir o Monitor Serial.

Abra um navegador e digite o endereço IP da placa para ter acesso ao servidor web. Clique nos botões e verifique no Monitor Serial se tudo parece estar funcionando como esperado.



```
0
Right
0
Right
10
Left
20
Left
30
Left
40
Left
50
Left
60
```

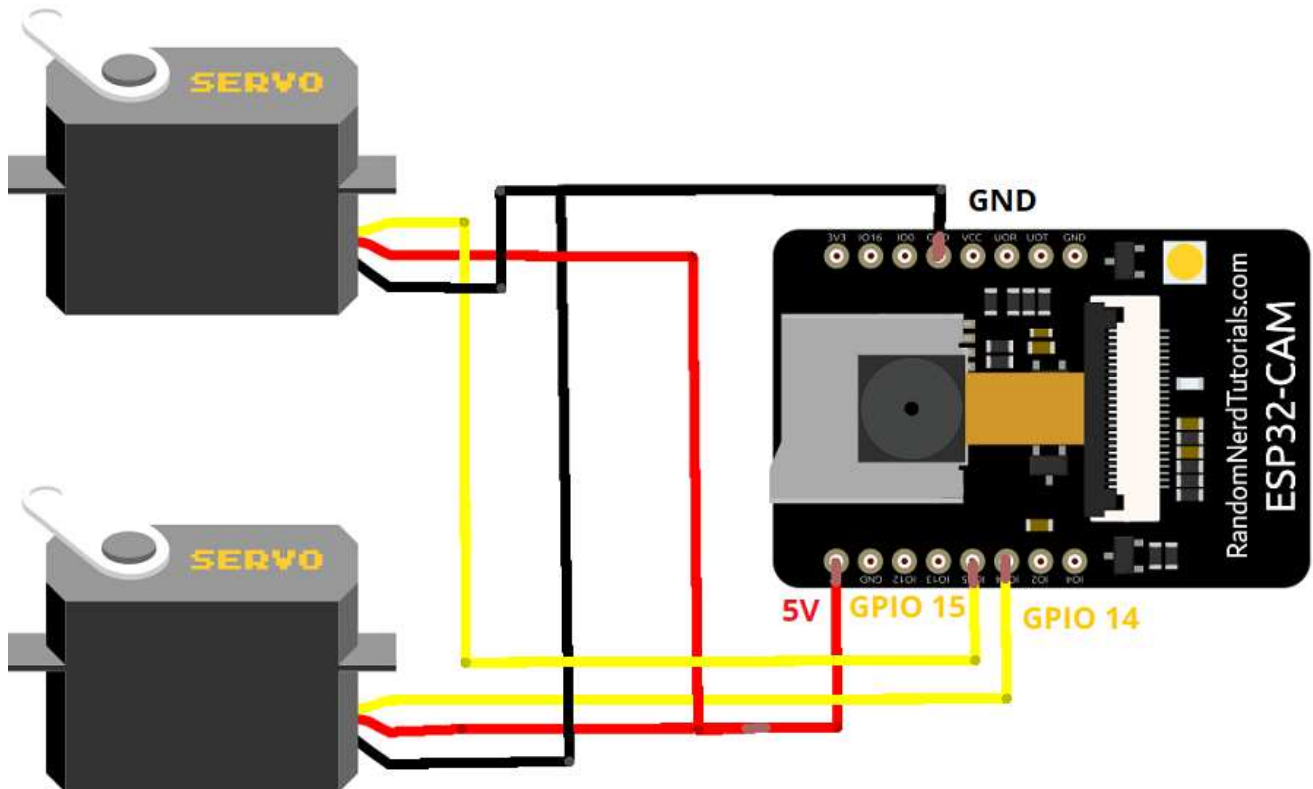
COM4

☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

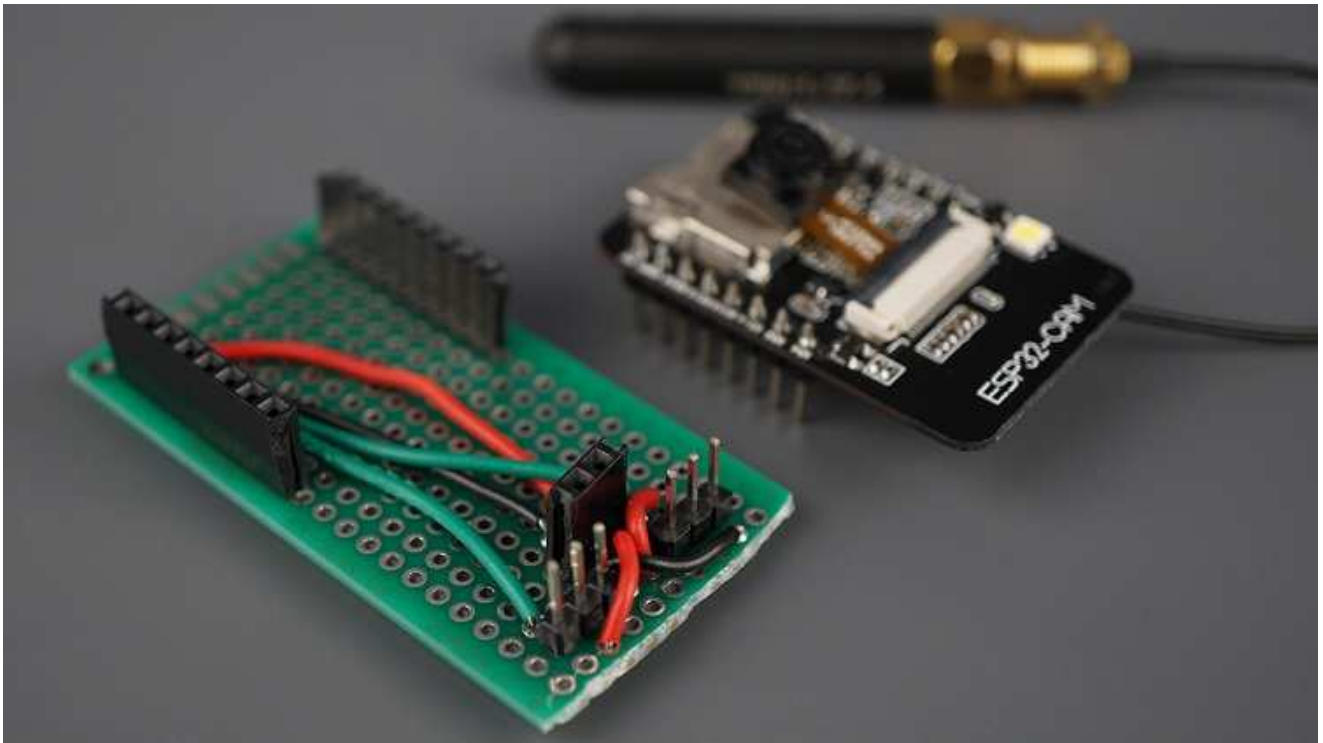
Se tudo estiver funcionando como esperado, você pode ligar os servomotores para o ESP32-CAM e continuar com o projeto.

Circuito

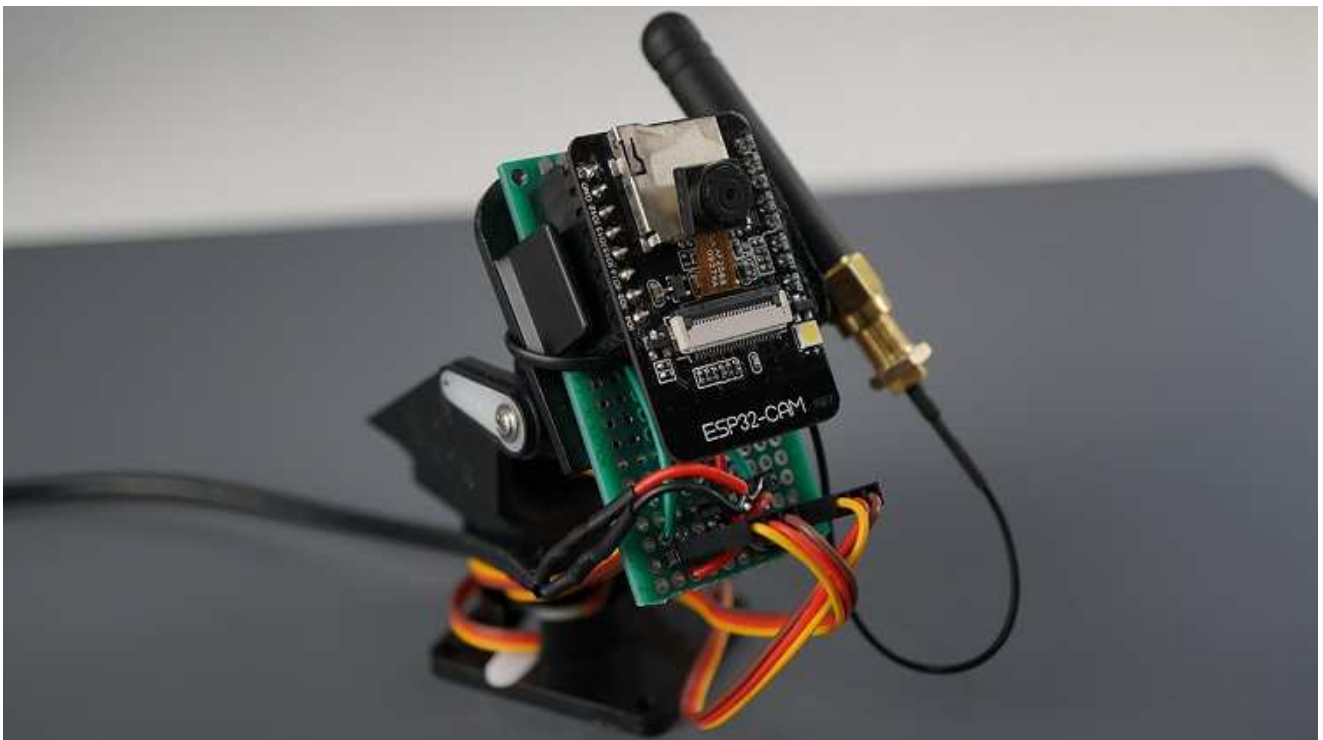
Depois de montar a panela e inclinar o suporte, conecte os servomotores ao ESP32-CAM, conforme mostrado no diagrama esquemático a seguir. Estamos conectando os pinos de dados do servo motor para **GPIO 15** e **GPIO 14**.



Você pode usar uma mini prancha para montar o circuito ou construir uma mini placa de popa com pinos de cabeçalho para conectar energia, o ESP32-CAM e os motores, como mostrado abaixo.



A figura a seguir mostra como a panela e as arquibancadas de inclinação se parecem após a montagem.



Demonstração

Aplique energia em sua prancha. Abra um navegador e digite o endereço IP ESP32-CAM. Uma página web com transmissão de vídeo em tempo real deve ser carregada. Clique nos botões para mover a câmera para cima, para baixo, para a esquerda ou para a direita.



Você pode mover a câmera remotamente usando os botões na página da Web. Isso permite que você monitore uma área diferente de acordo com a posição da câmera. Esta é uma ótima solução para aplicações de vigilância.

Embrulhando

Neste tutorial, você aprendeu a construir um servidor web pan e tilt com streaming de vídeo para controlar o ESP32-CAM.

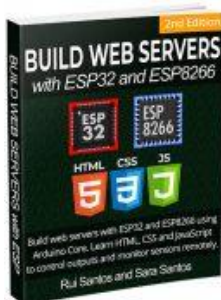
Controlar os servomotores com o ESP32-CAM é o mesmo que controlá-los usando um ESP32 "regular". Você pode ler o tutorial a seguir para saber mais sobre servomotores com o ESP32:

- [ESP32 Servo Motor Web Server com Arduino IDE](https://randomnerdtutorials.com/esp32-cam-pan-and-tilt-2-axis/)

Se você quiser controlar seu robô fora do alcance de sua rede local, você pode considerar [definir o ESP32-CAM como um ponto de acesso](#). Desta forma, o ESP32-CAM não precisa se conectar ao seu roteador. Ele cria sua própria rede wi-fi, e dispositivos wi-fi próximos como seu smartphone podem se conectar a ele.

Para mais projetos e tutoriais com o ESP32-CAM:

- [Construa projetos ESP32-CAM usando eBook Arduino IDE](#)
- [Mais projetos e tutoriais ESP32-CAM...](#)

A green and yellow banner for PCBWay. On the left, the PCBWay logo is at the top, followed by the text 'PCB Fabrication & Assembly'. Below this, 'ONLY \$5 for 10 PCBs' is written in large, bold letters. Underneath, there are two checkmarks: '24-hour Build Time' and 'Quality Guaranteed'. Below these, it says 'Most Soldermask Colors:' followed by a row of ten color swatches: green, red, yellow, blue, white, black, pink, grey, orange, and purple. At the bottom left is an orange button that says 'Order now'. On the right side of the banner is a circular inset image showing a complex PCB with various components, with the website 'www.pcbway.com' overlaid on it.

[eBook] Construir servidores web com ESP32 e ESP8266 (2ª Edição)

Build Web Server projects with the ESP32 and ESP8266 boards to control outputs and monitor sensors remotely. Learn HTML, CSS, JavaScript and client-server communication protocols [DOWNLOAD »](#)

Recursos recomendados