

## Inicialização e Captura de Imagem

```
1. #include "esp_camera.h"
2.
3. // Configuração dos pinos da câmera AI-Thinker
4. #define PWDN_GPIO_NUM    32
5. #define RESET_GPIO_NUM   -1
6. #define XCLK_GPIO_NUM     0
7. #define SIOD_GPIO_NUM    26
8. #define SIOC_GPIO_NUM    27
9. #define Y9_GPIO_NUM       35
10. #define Y8_GPIO_NUM       34
11. #define Y7_GPIO_NUM       39
12. #define Y6_GPIO_NUM       36
13. #define Y5_GPIO_NUM       21
14. #define Y4_GPIO_NUM       19
15. #define Y3_GPIO_NUM       18
16. #define Y2_GPIO_NUM        5
17. #define VSYNC_GPIO_NUM    25
18. #define HREF_GPIO_NUM     23
19. #define PCLK_GPIO_NUM     22
20.
21. // Configuração da câmera
22. camera_config_t camera_config = {
23.     .pin_pwdn = PWDN_GPIO_NUM,
24.     .pin_reset = RESET_GPIO_NUM,
25.     .pin_xclk = XCLK_GPIO_NUM,
26.     .pin_sscb_sda = SIOD_GPIO_NUM,
27.     .pin_sscb_scl = SIOC_GPIO_NUM,
28.     .pin_d7 = Y9_GPIO_NUM,
29.     .pin_d6 = Y8_GPIO_NUM,
30.     .pin_d5 = Y7_GPIO_NUM,
31.     .pin_d4 = Y6_GPIO_NUM,
32.     .pin_d3 = Y5_GPIO_NUM,
33.     .pin_d2 = Y4_GPIO_NUM,
34.     .pin_d1 = Y3_GPIO_NUM,
35.     .pin_d0 = Y2_GPIO_NUM,
36.     .pin_vsync = VSYNC_GPIO_NUM,
37.     .pin_href = HREF_GPIO_NUM,
38.     .pin_pclk = PCLK_GPIO_NUM,
39.     .xclk_freq_hz = 20000000,
40.     .ledc_timer = LEDC_TIMER_0,
41.     .ledc_channel = LEDC_CHANNEL_0,
42.     .pixel_format = PIXFORMAT_JPEG, // Formato da imagem
43.     .frame_size = FRAMESIZE_QVGA,   // Resolução da imagem
44.     .jpeg_quality = 10,              // Qualidade JPEG
45.     .fb_count = 1,                  // Número de frame buffers
46. };
47.
48. void setup() {
49.     Serial.begin(115200); // Inicializa o monitor serial
50.
51.     // Inicializa a câmera
52.     if (esp_camera_init(&camera_config) != ESP_OK) {
53.         Serial.println("Falha ao inicializar a câmera!");
54.         while (true); // Para o código em caso de erro
55.     }
56.     Serial.println("Câmera inicializada com sucesso!");
57. }
```

```

58.
59. void loop() {
60.     // Captura uma imagem
61.     camera_fb_t *fb = esp_camera_fb_get();
62.     if (!fb) {
63.         Serial.println("Falha ao capturar imagem!");
64.         return;
65.     }
66.
67.     // Exibe o tamanho da imagem capturada no monitor serial
68.     Serial.printf("Imagem capturada! Tamanho: %u bytes\n", fb->len);
69.
70.     // Libera o frame buffer
71.     esp_camera_fb_return(fb);
72.
73.     delay(5000); // Aguarda 5 segundos antes de capturar novamente
74. }
75.
76.

```

## Funcionalidade do Código

### Inicialização da Câmera:

Configura os pinos GPIO e os parâmetros da câmera (resolução, qualidade, etc.).

Usa `esp_camera_init` para inicializar o hardware da câmera.

### Captura de Imagem:

Usa `esp_camera_fb_get` para capturar uma imagem e armazená-la no buffer.

O tamanho do arquivo capturado (em bytes) é exibido no monitor serial.

O buffer é liberado após o uso com `esp_camera_fb_return`.

### Repetição Contínua:

O `loop` captura imagens a cada 5 segundos, podendo ser ajustado para outros intervalos.

### Configuração de Hardware:

Os pinos da câmera devem estar corretamente conectados ao ESP32-CAM.