# ESP32-S3 CAM Color-Tracking Pan-Tilt System

A real-time computer-vision tracker built with the ESP32-S3 CAM board and two MG90S/SG90 servos.  
  
This project uses the ESP32-S3 CAM and Arduino IDE to detect a specific colour in the camera frame and automatically move a pan-tilt mount to keep that object centred in view — no external computer required.

## Features

* Real-time colour tracking using onboard camera
* 2-axis pan-tilt control with MG90S/SG90 micro servos
* Simple proportional (P-controller) centring algorithm
* Fully self-contained — no OpenMV, Pi, or PC processing
* Optional SD-card support for logging or future expansion

## Bill of Materials

|  |  |  |
| --- | --- | --- |
| Component | Quantity | Notes |
| ESP32-S3 CAM (16 MB, OV2640/OV5640) | 1 | Main processor and camera |
| FTDI USB-to-TTL Programmer | 1 | For code upload (3.3 V/5 V switchable) |
| MG90S or SG90 Micro Servos | 2 | One for pan, one for tilt |
| Pan-Tilt Bracket | 1 | Acrylic or 3D-printed |
| 100 µF Electrolytic Capacitor | 1 | Power rail smoothing |
| 100 nF Ceramic Capacitor | 2 | Near each servo for noise reduction |
| Jumper Wires | - | Male/female as required |
| Breadboard | - | For connections |
| 5 V Bench Power Supply | 1 | Shared servo and ESP32 power |
| MicroSD Card (optional) | 1 | For recording/logging |

## Wiring Overview

Pan Servo Signal → GPIO 14  
Tilt Servo Signal → GPIO 15  
Servo VCC → 5 V from PSU (not from ESP32 pin)  
Servo GND → Common with ESP32 GND  
FTDI TX → ESP32 RX (U0R)  
FTDI RX → ESP32 TX (U0T)  
FTDI 5 V → ESP32 5 V, FTDI GND → ESP32 GND  
IO0 → GND (only during upload), then disconnect and reset to run.  
  
Add the 100 µF capacitor across the 5 V ↔ GND near the servos.

## Software Setup

1. Install Arduino IDE (v2.2+ recommended).
2. In Boards Manager, install: esp32 by Espressif Systems.
3. Select board: ESP32-S3 Dev Module.
4. Install library: ServoESP32.
5. Copy the provided sketch into Arduino IDE.
6. Connect FTDI → ESP32-S3-CAM (as above).
7. Hold IO0 → GND, press RESET, then upload.
8. Remove IO0 from GND and press RESET to start.

## Color Detection

Inside the code, tune these thresholds to match your object:  
  
uint8\_t targetHueMin = 5;  
uint8\_t targetHueMax = 25;  
uint8\_t targetSatMin = 100;  
uint8\_t targetValMin = 60;  
  
These values track orange by default. To detect another colour, use an HSV colour picker and substitute hue/saturation/value ranges accordingly.

## Operation

1. Power the system (5 V / 2 A recommended).
2. The servos initialise to centre (90°, 90°).
3. When the target colour appears in view, the ESP32 calculates its centroid.
4. The pan/tilt servos move to re-centre the object.
5. Serial Monitor (115200 baud) shows tracking info.

## Fine-Tuning

Adjust Kp (proportional gain) for faster or smoother motion:  
float Kp = 0.05;  
  
If the image is noisy, increase the pixel sampling step in code.  
Use FRAMESIZE\_QVGA for ~10 fps performance.  
Ensure good, even lighting for accurate HSV detection.

## Future Expansions

* Add Wi-Fi streaming to visualise tracking.
* Record servo motion to SD for analysis.
* Upgrade to full object detection using TensorFlow Lite on the ESP32-S3.
* Integrate with ultrasonic or IR sensors for distance feedback.

## License

Open-source under the MIT License. Use and modify freely for educational or hobby projects.