A black background with grey leaves

AI-generated content may be incorrect.

Iot color detection control board

Designed by abbas ghadiri

A grey logo on a black background

AI-generated content may be incorrect.

May 18, 2025

Abstract:

This project involves the design of an IoT board based on the ESP32-S microcontroller to detect specific color ranges through a camera module. The detected color data is processed and sent to a remote server using the HTTP POST method in a JSON format.

Requirements:

* Hardware:
* ESP32-s (or ESP32CAM)
* OV2640 camera
* FFC 24 pin connector and cable (right cable and bottom connector)
* 5V to 3.3V AMS117 regulator
* 4-pin phoenix connector
* Micro USB SMD 4-pin-holder
* Relay 5V
* Buzzer 5V
* IPUS PSRAM
* CH340G
* 12.000MHZ oscillator
* TAC switch.
* Software:
* VScode, platform IO extension.
* Arduino IDE
* Python (or PHP)
* XAMPP (or APACHE and MYSQL)

Introduction

Color detection systems are widely used in industrial, agricultural, and consumer applications. This project aims to implement a low-cost, embedded color detection system that transmits data to a server for monitoring or control purposes. The system is based on ESP32 and can be integrated with other IoT systems.

**System Overview**

The system includes a camera (OV2640) that captures an image, a ESP32-S microcontroller that processes the image to detect color ranges and sends data to a server. The system also includes output modules such as a relay or buzzer that can be triggered based on the color detection result.

**Hardware Architecture**

The ESP32-S is used as the main controller due to its processing power and built-in WIFI capability. The OV2640 camera captures images, and additional components like a relay and buzzer provide physical output. The AMS117 regulator ensures proper voltage conversion, and the CH340G allows USB communication.

A schematic diagram should include:

* ESP32 connections
* OV2640 wiring
* Relay and buzzer control lines
* Power supply architecture

**Software Architecture**

The ESP32 is programmed using the Arduino framework with Platform-IO. The program follows this sequence:

1. Initialize camera and WIFI
2. Capture image
3. Process image to detect color using RGB/HSV thresholding
4. Create JSON object with color detection result
5. Send JSON to server via HTTP POST

**Server Side**

The server receives POST requests using a PHP file. The data can be stored in a MySQL database using XAMPP. Server-side code parses the JSON and logs the information.

**Applications**

* Industrial color-based sorting
* Smart home automation (e.g., change lighting based on object color)
* Educational projects

**Future Improvements**

* Implement AI-based image recognition for better accuracy
* Add wireless communication protocols (MQTT, BLE)
* Mobile app integration

**Appendix**

* Full BOM (Bill of Materials links)
* Source code link [(GitHub repository)](https://github.com/abbasghadir/IoT_color_detection/tree/main/software)
* Schematic diagram and PCB layout [(GitHub repository)](https://github.com/abbasghadir/IoT_color_detection/tree/main/hardware)