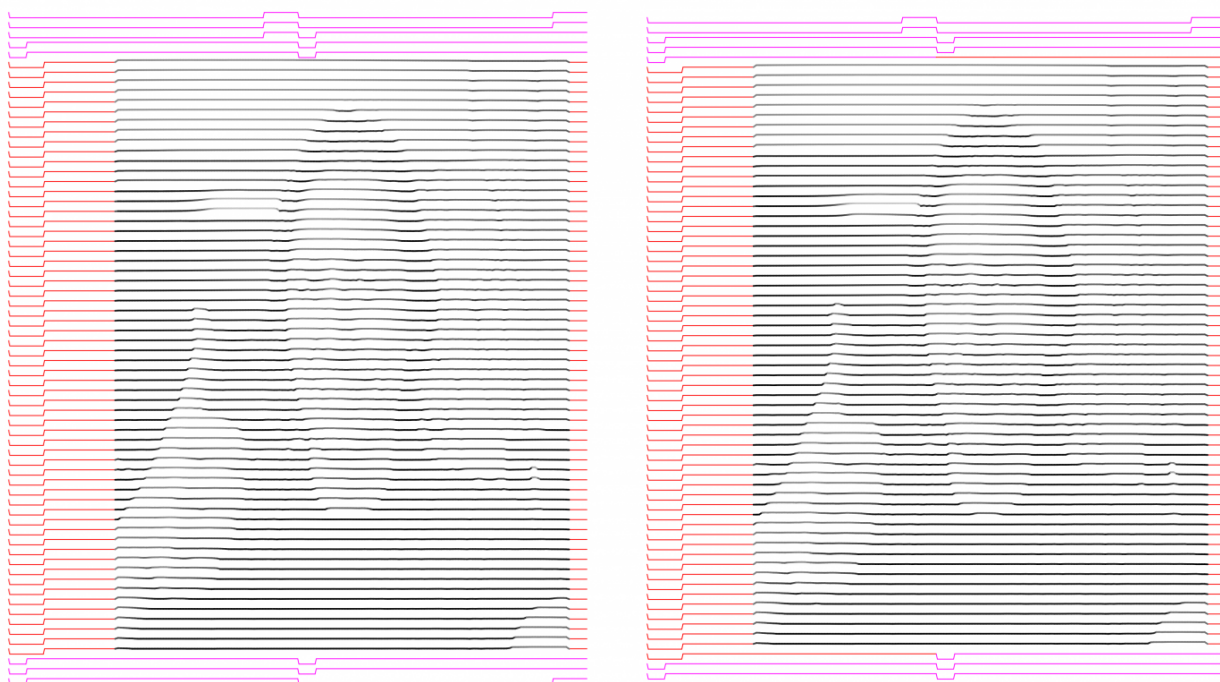


Nguyên tắc làm việc

Mã tạo ra tín hiệu tương tự bằng cách sử dụng DAC bên trong theo tiêu chuẩn PAL và NTSC được mô tả [ở đây](#).

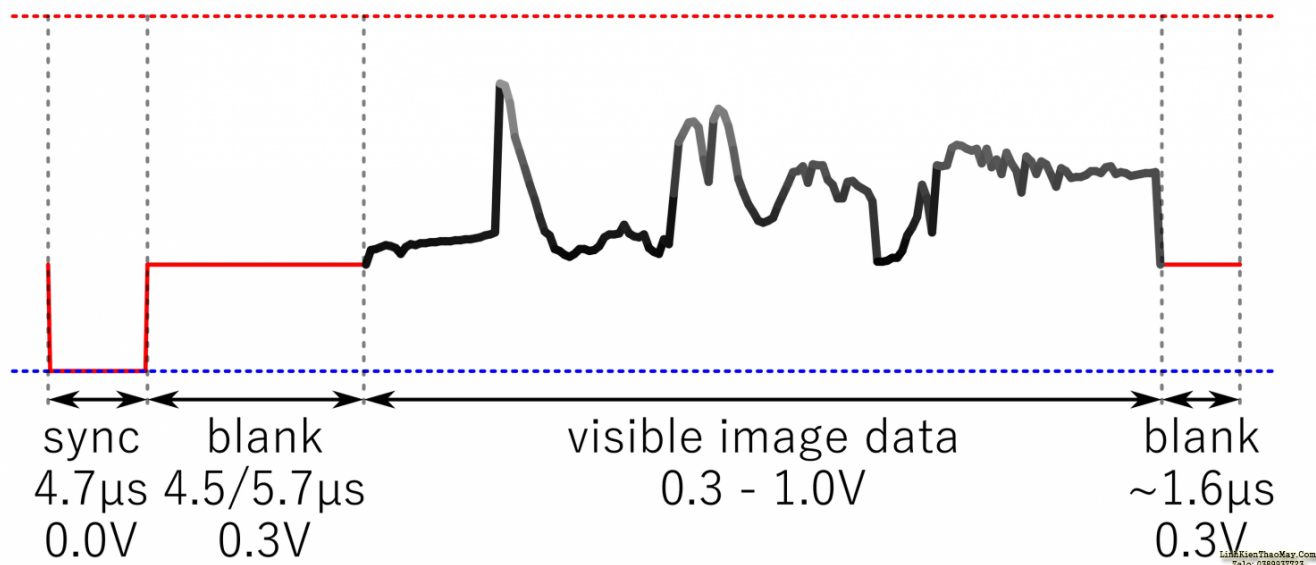
Mỗi hình ảnh bao gồm hai nửa khung hình (dòng hình ảnh chẵn và lẻ). Mức trống ở mức $\sim 0,3V$ (giá trị phụ thuộc vào PAL/NTSC). Việc đồng bộ hóa được thực hiện bằng cách sử dụng một số xung ở mức $0,0V$.

Hình ảnh sau đây cung cấp một cái nhìn tổng quan. Đồng bộ hóa theo chiều dọc đạt được bằng các xung màu tím, theo chiều ngang bằng các xung màu đỏ:

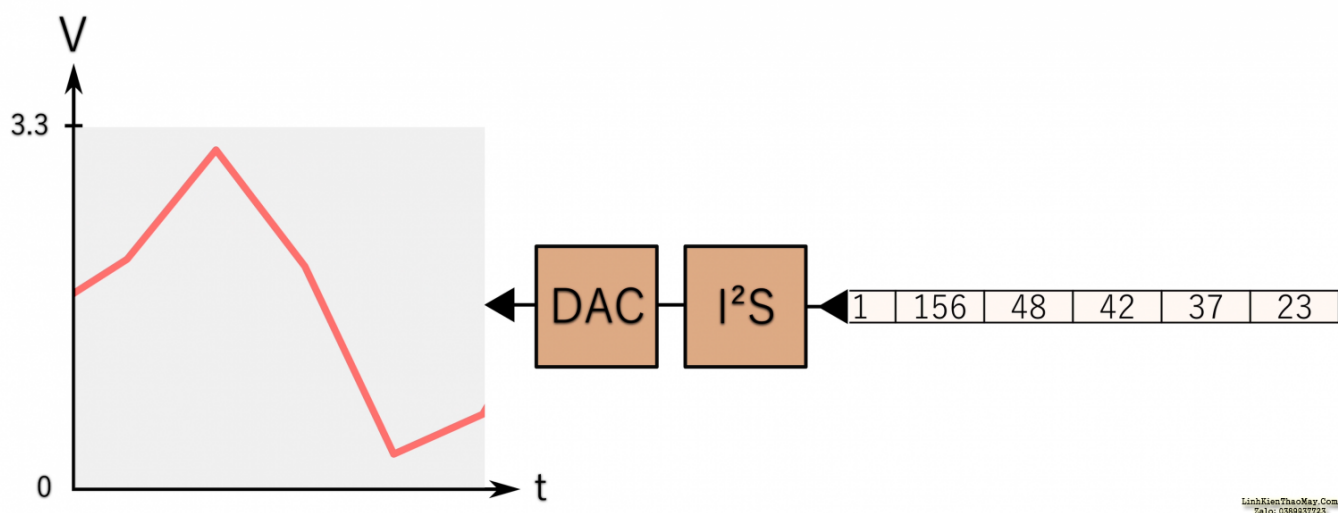


LinhKienThaoMay.Com
Zalo: 0389937723

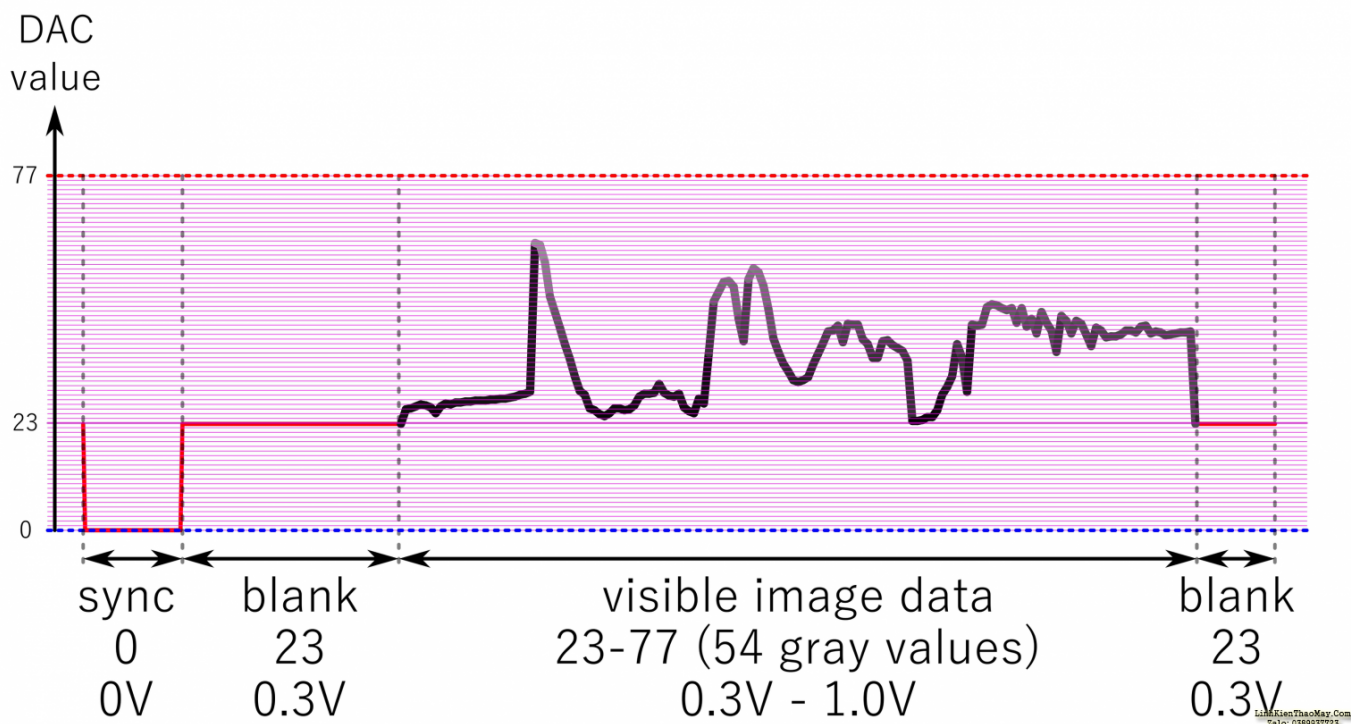
Hình ảnh này cung cấp cái nhìn tổng quan về tín hiệu của một đường hình ảnh (tổng cộng là $64\mu s$):



Để tạo tín hiệu, các DAC 8 bit bên trong được sử dụng có thể tạo ra các mức điện áp trong khoảng từ 0 đến 3,3V với tốc độ cực nhanh bằng I2S.



Vì tín hiệu tổng hợp được xác định trong khoảng từ 0 đến 1V nên chỉ các giá trị DAC từ 0 đến 77 là hữu ích. Điều này dẫn đến 54 giá trị màu xám cho hình ảnh.



Màu sắc hiện chưa được triển khai vì nó dựa trên tần số trên 4 MHz. mình có thể tạo ra một tín hiệu đơn giản có tần số 6,66 MHz chỉ bao gồm hai giá trị, nhưng mình sẽ bỏ lỡ tần số mục tiêu và sự dịch pha sẽ dẫn đến các màu ngẫu nhiên. Tuy nhiên điều này có thể được khám phá trong một Project trong tương lai.

Các linh kiện

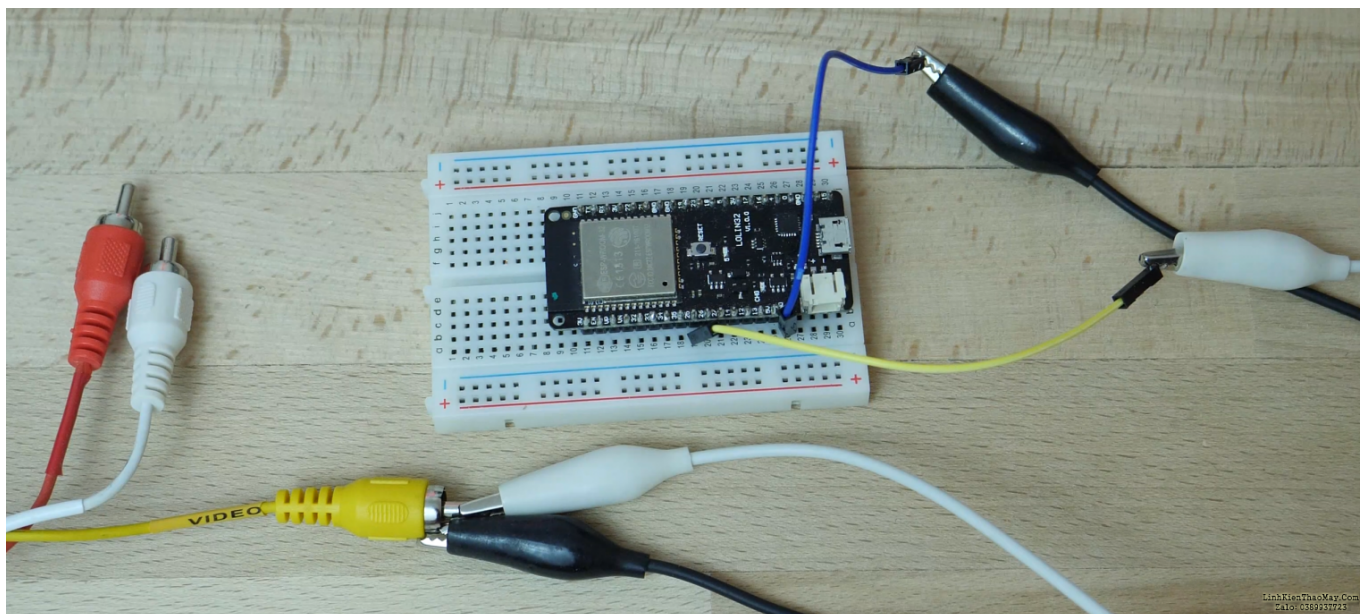
linh kiện được sử dụng ở đây là bo mạch LOLIN32. Nhưng các bảng ESP32 nào cũng có thể được sử dụng.



Cài đặt

Việc thiết lập khá đơn giản. **Chân 25** phải được nối **vào bên trong đầu** nối cinch màu vàng và **nối đất với** phần bên ngoài. Chỉ như thế này:

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723



Mã số

Mã được viết bằng C++ và chạy trong Arduino IDE. Cần tích hợp Arduino ESP32 để biên dịch mã (Vui lòng làm theo hướng dẫn tại đây: <https://github.com/espressif/arduino-esp32>).

Tất cả các mã có thể được tìm thấy ở đây. README mô tả nội dung kho lưu trữ:
<https://github.com/bitluni/ESP32CompositeVideo>

Nếu bạn tìm thấy lỗi hoặc có một số cải tiến, chỉ cần gửi yêu cầu kéo trên github... mình thực sự đã đọc tin nhắn của mình ở đó 😊

Trình chuyển đổi hình ảnh thành tiêu đề

Lớp Hình ảnh cho phép bạn hiển thị hình ảnh thang màu xám trên màn hình và các lớp Đồ họa và Phong chữ cho phép in văn bản.

Tuy nhiên, giá trị hình ảnh không được vượt quá giá trị mức xám được hỗ trợ tối đa hiện là 54. Giá trị này phải được đặt trong bộ chuyển đổi trước khi chọn hình ảnh. Giá trị cao hơn có thể được sử dụng làm chỉ số minh bạch. Phong chữ là Hình ảnh của các ký tự ASCII bắt đầu từ 32. Kiểm tra png của phong chữ trong kho lưu trữ. Các phong chữ có thể được chuyển đổi bằng cách sử dụng giá trị tối đa 255. Hiện tại, 0 được sử dụng làm nền và các thứ gì làm pixel nền trước.

<https://bitluni.net/wp-content/uploads/2018/02/Image2Header.html>

Các hình ảnh đã chuyển đổi có thể được đặt trong thư mục Project và chỉ cần đưa vào Project. như thế này: `#include "luni.h" Hình ảnh<CompositeGraphics> luni0(luni::xres, luni::yres, luni::pixels); ... { đồ họa.begin(0); //xóa màn hình đen luni0.draw(đồ họa, 200, 10); đồ họa.end ();`

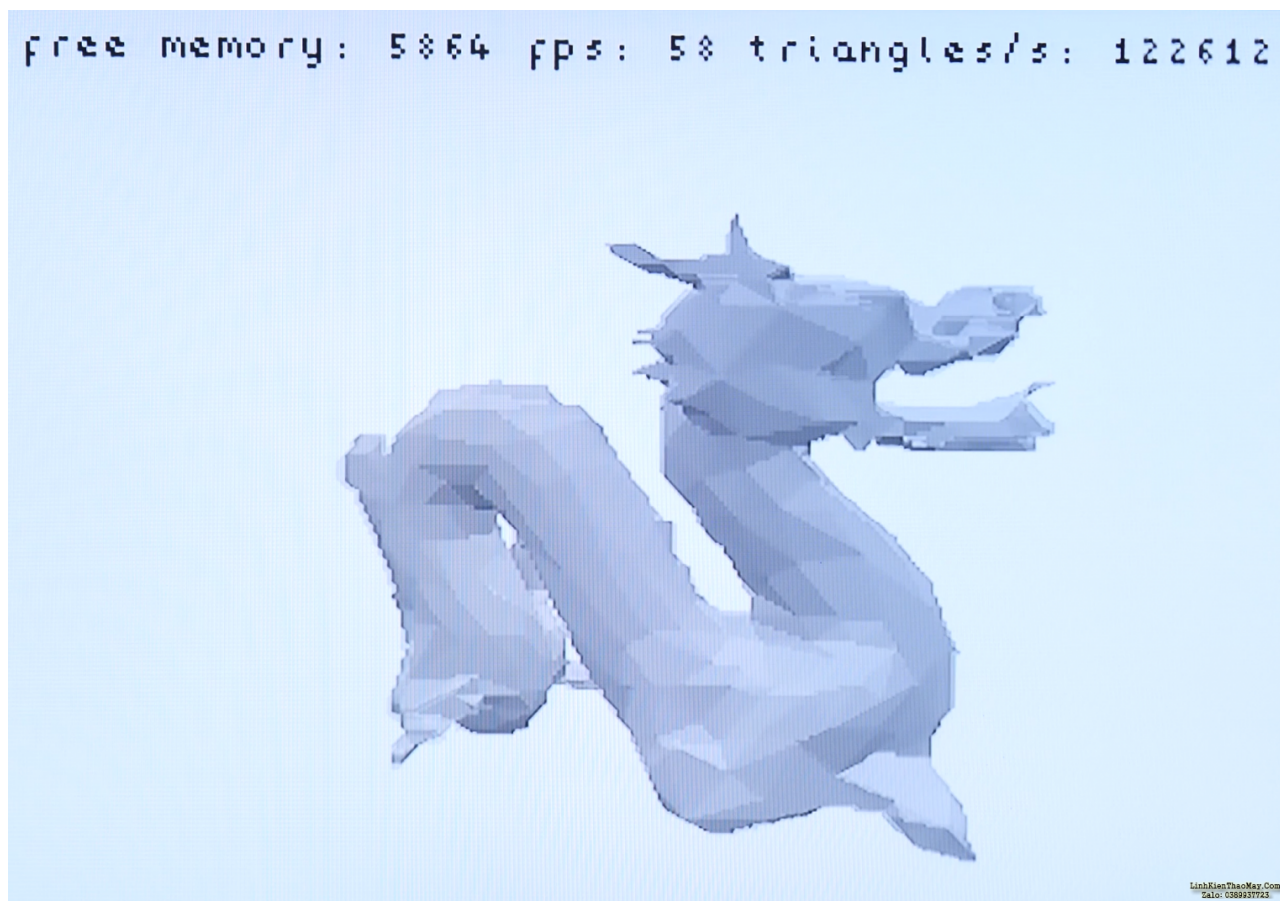
Kiểm tra Project [CompositeVideoSimple](#) từ kho lưu trữ github.

Công cụ 3D

Đáng tiếc là Esp32 không có nhiều bộ nhớ. Phần lớn được sử dụng cho vùng đệm trước và sau của hình ảnh. Vì vậy, không còn chỗ cho Z-Buffer ở độ phân giải hợp lý.

Như một giải pháp thay thế, công cụ này thu thập các hình tam giác trong một cây được sắp xếp theo độ sâu và được hiển thị ở cuối. Bộ đệm cho cây chiếm ít không gian hơn nhưng có khả năng tính toán cao hơn. Kích thước của bộ đệm này phải được đặt bằng tham số thứ ba khi tạo đối tượng đồ họa (hiện tại): Đồ họa CompositeGraphics(XRES, YRES, 1337);

Tham số mô tả số lượng hình tam giác tối đa sẽ được vẽ. Kể từ khi mặt sau loại bỏ các mặt lưới được thực hiện khoảng. gấp đôi số lượng tam giác có thể được rút ra. Bản thân lưới cần thêm không gian cho các đỉnh được chuyển đổi. mình có thể vẽ tổng số mô hình khoảng 2k hình tam giác ở độ phân giải CGA (320×200), độ phân giải này đã lấp đầy gần như toàn bộ 200kb có sẵn khi bắt đầu.



Bộ chuyển đổi tiêu đề STL sang tiêu đề

Phiên bản mới này của trình chuyển đổi stl bao bọc dữ liệu đã xuất trong một không gian tên được đặt tên theo tệp nguồn.

TRUNG TÂM SỬA CHỮA ĐIỆN TỬ QUẢNG BÌNH

MR. XÔ - 0901.679.359 - 80 Võ Thị Sáu, Phường Quảng Thuận, tx Ba Đồn, tỉnh Quảng Bình

GIÁ RẺ

NHANH CHÓNG

LINH KIỆN CHÍNH HÃNG



**TRUNG TÂM SỬA CHỮA ĐIỆN TỬ
XÔ NGUYỄN**

- Dịch vụ sửa chữa điện tử tại nhà
- Cung cấp linh kiện điện tử
- Tư vấn lắp đặt nhà thông minh

Đc: Quảng Thuận, tx Ba Đồn,
tỉnh Quảng Bình - 0901.679.359

<https://bitluni.net/wp-content/uploads/2018/02/StlConverter2.html>

Các mắt lưới đã chuyển đổi có thể được đặt trong thư mục Project và được bao gồm như thế

Tài liệu này được tải từ website: <http://linhkienthaomay.com>. Zalo hỗ trợ: 0389937723

này: #include "venus.h" Mô hình Mesh<CompositeGraphics>(venus::vertexCount, venus::vertices, 0, 0, venus::triangleCount, venus::triangles, venus::triangleNormals); ... {
đồ họa.begin(54); //xóa màn hình trắng model.transform(transformationMatrix, spinningMatrix); //chuyển đổi các đỉnh và xoay pháp tuyến model.drawTriangles(đồ họa, 40); // xếp hàng các tirangles và sắp xếp chúng đồ họa.flush (); //vẽ các tam giác đã được sắp xếp đồ họa.end ();

Các bài viết tương tự:

1. [DIY Camera ESP32 I2S \(OV7670\)](#)
2. [foxcom g31 - chet bios](#)
3. [fujitsu nhật bãi 110v 9000btu inveter gas 410 - làm lạnh nóng bình thường lộc kê to](#)
4. [Hướng dẫn dùng ESP32 xuất hình lên máy hiện sóng](#)
5. [LCD led DELL S2240L - e cần mua Panel](#)
6. [Monitor CRL LG 17' - Co đen xanh nhưng khong len, mo ra thay phong cao ap](#)
7. [nguồn - bat nguồn quat khong quay may khong khoi dong](#)
8. [nhạc sống cho các bác yêu nhạc sống - ngày mới việc mới kinh nghiệm mới,,,sảng khoái](#)
9. [phan cứng - main pc, laptop](#)
10. [Sam Sung 931BW - Màn chớp tắt](#)
11. [Thùng loa di động - vặn volume nhỏ bị nghẹt tiếng](#)
12. [tivi lcd lg 32ld310 - máy chấp ic nguồn w 6053 và d503,504](#)