ESPHOME IR Sender

Send IR signals through ESPHOME, to recreate any IR controlled device

- HDMI Switcher IR Controller

HDMI Switcher IR Controller

This project allows you to control an HDMI switch (or any IR controlled device) using an **ESP32** running ESPHome.
It also provides a way to **capture IR codes with an Arduino Uno R3**, so you can easily copy them into the ESPHome configuration.

The goal is to make this setup open-source, easy to reproduce, and fully documented.

Table of Contents

Hardware Requirements

ESP32 Side (IR Transmitter):

- ESP32 Dev Board (ESP32-DevKitC or similar)
- IR LED connected to **GPIO13**
- ESPHhome already installed

image

Note: Check the datasheet of your IR LED to see what pins will be connected to GND and GPIO13

Arduino Side (IR Receiver for Capturing Codes):

- Arduino Uno R3
- IR Receiver module connected to **Pin 4**

Screenshot 2026-02-21 102123

Note: Check the datasheet of your IR Reciver to see what pins will be connected to GND and GPIO4

ESPHome Setup

The ESP32 runs ESPHome to transmit IR codes to the HDMI switch.

How to Use

1. Copy `hdmi-switcher-ir.yaml` (Download) to your ESPHome folder.
2. Fill in the placeholders for:
   - Wi-Fi credentials (Change in ESPHOME settings)
   - OTA password (Line 20)
   - API encryption key (Line 16)
   - Raw IR codes for your HDMI inputs

   > **Note:** To find the OTA and API key just make a blank yaml config in ESPHOME

3. Upload the YAML to the ESP32 via OTA or USB.

   > **Note:** The raw IR codes can be captured using the Arduino setup below. **Note:** The current IR signals are configured for a UGREEN HDMI Switcher

---

Arduino IR Capture

The Arduino is used to read raw IR signals from your HDMI remote and format them for ESPHome.

How to Use

1. Connect your IR receiver to Arduino Pin 4.
2. Upload `ir_capture.ino` (Download) to the Arduino.
3. Open the Serial Monitor at **115200 baud**.
4. Press a button on your HDMI remote.
5. Copy the array output from the Serial Monitor. This is the exact format needed for `transmit_raw` in ESPHome.

Capturing and Adding New IR Codes

1. Run the Arduino IR capture sketch.
2. Press the desired button on your HDMI remote.
3. Copy the outputted array (e.g., `[8850, -4500, 550, -550, ...]`).
4. Paste it into the corresponding button in `hdmi-switcher-ir.yaml`.
5. Save and upload your updated YAML to the ESP32.
6. Make sure to refer to IRremote Arduino Library to add new features

---

Web Interface & Buttons

The ESPHome web server allows you to:

- Press buttons from a browser
- Test IR codes without integrating into Home Assistant

All buttons defined in `hdmi-switcher-ir.yaml` will appear in the web interface. And you should be able to access the device web ui from hdmi-switcher-ir.local (or whatever you rename the yaml)

Tips & Troubleshooting

- **IR not working:** Try sending the signal multiple times using `repeat:` in ESPHome.

- **Carrier mismatch:** Some devices prefer 36–40 kHz. Adjust `carrier_frequency` if needed.
- **Series resistor:** Protect your IR LED with a 100–220Ω resistor.
- **Static IP:** Optional but helpful for accessing the ESP32 consistently via web UI.
- **Debugging:** Use the ESPHome logger to confirm button presses are being triggered.

Case

Additonally there is a case available on nexprint (coming soon)

File Structure

```
├── README.md # This file
├── hdmi-switcher-ir.yaml # ESPHome YAML config
└── ir_capture.ino # Arduino sketch to capture IR codes
```

License

References

- ESPHome Remote Transmitter Docs
- IRremote Arduino Library