

ESP32 Infrared Remote

Metadata

- **Author:** Petru Zincenco
- **Master:** IA
- **Date Completed:** 21 May 2025

Introduction

This project extends an existing Home Assistant setup (on Raspberry Pi 4) to control IR-only devices via an ESP32 acting as an IR transceiver, and to monitor indoor temperature and humidity. We use the YS-IRTM module to sniff and replay NEC-protocol commands, and a DHT11 sensor to read environment data every 30 minutes. All IR commands and sensor readings are exposed in Home Assistant as entities.

General Description

Block Diagram



Modules and Interactions

- **Home Assistant Server** (Raspberry Pi 4)
 1. Runs ESPHome add-on for firmware management, API, MQTT and OTA.
 2. Receives IR command events and sensor data from ESP32, exposes them as switches and sensors.
- **ESP32 IR & Sensor Hub**
 1. Connects over Wi-Fi (static IP) to Home Assistant via ESPHome native API.
 2. Interfaces via UART (irSerial.begin at 9600, SERIAL_8N1) on GPIO 22 (RX) and GPIO 23 (TX) with YS-IRTM.
 3. Reads DHT11 sensor on GPIO 21 every 30 minutes.
- **YS-IRTM IR Emitter/Receiver Module**
 1. 38 kHz carrier, hardware-decode for NEC frames only.
 2. Outputs 3-byte payloads on receive; accepts 5-byte payloads to transmit.
- **DHT11 Temperature & Humidity Sensor**
 1. Reads temperature and humidity on GPIO 21; data logged every 30 minutes.

Hardware Design

Parts List

- ESP32 development board - 46 RON



- YS-IRTM IR emitter/receiver module (38 kHz, NEC) - 22 RON



- DHT11 sensor module - 15 RON



- 4-channel 3.3 V ↔ 5 V level shifter - 11 RON



- Jumper wires and breadboard

Total cost: ≈ 94 RON

Connections and Wiring

- **YS-IRTM Module**

1. LVCC → 3.3 V (ESP32)
2. HVCC → 5 V (ESP32)
3. GND → ESP32 GND
4. TXD → level shifter → ESP32 GPIO 22 (RX)
5. RXD ← level shifter ← ESP32 GPIO 23 (TX)

- **DHT11 Sensor**

1. VCC → 5 V (ESP32)
2. GND → ESP32 GND
3. DATA → ESP32 GPIO 21

- **IR LED & Driver**

1. IR LED anode → 5 V via resistor → transistor → ESP32 GPIO (configured by module)
2. Common GND for all devices

Software Design

Environment & Components

- **ESPHome** (YAML-based) for Wi-Fi, API, MQTT, OTA.
- ``uart`` on GPIO 22/23 at 9600 baud, SERIAL_8N1.
- ``remote_receiver`` + ``remote_transmitter`` for YS-IRTM NEC frames.
- ``dht`` sensor component on GPIO 21, `update_interval: 1800 s`.
- **Custom ``text_sensor``** lambda to parse 3-byte NEC payloads into HA events.

Receive Handling

1. YS-IRTM decodes NEC IR and emits 3 bytes over UART.
2. ESPHome buffers 3 bytes, then publishes an MQTT/state update (``ir_hub/last_command``).
3. Home Assistant sensor template splits into user-code and command.

Transmit Handling

1. HA invokes ``switch.ir_send_<device>_<cmd>``.
2. ESPHome writes 5-byte payload (Addr: 0xA1, F1: 0xF1, UC High, UC Low, Cmd) to UART.
3. YS-IRTM emits NEC frame (9 ms mark, 4.5 ms space, data bits 562 μs @38 kHz).
4. Module replies 0xF1 on success; ESPHome retries once after 1 s if no ack.

Sensor Handling

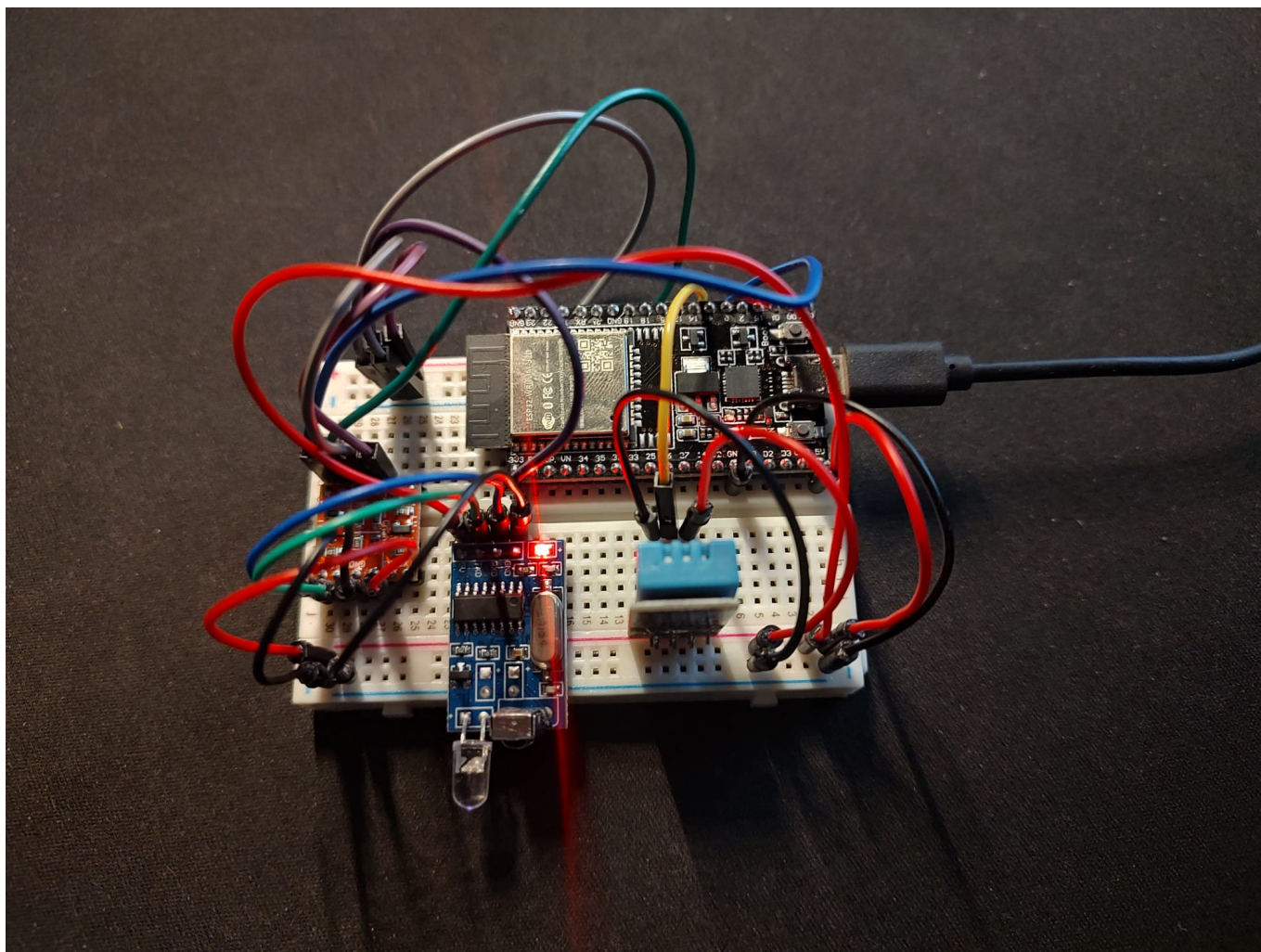
- DHT11 on GPIO 21 is read every 30 minutes.
- Temperature and humidity values are published to Home Assistant as ``sensor.inside_temperature`` and ``sensor.inside_humidity``.

Known IR Codes

Function	NEC Code
Power On/Off	01 FE 04
Mute	01 FE 09
Volume Down	01 FE 05
Volume Up	01 FE 06

Images
 Images

Hardware setup: ESP32, YS-IRTM module and DHT11 sensor wired on the breadboard and powered on.



ESP32 in Home Assistant UI: The ESP32 entity and exposed IR switches and sensor values displayed on the HA dashboard.



Inside Temperature Graph: Logged DHT11 readings showing temperature variations over time.



Inside Humidity Graph: Logged DHT11 readings showing humidity fluctuations over time.



Results

All NEC remotes in the stereo system are learned and replayed via Home Assistant. Indoor temperature and humidity are logged every 30 minutes and available in HA dashboards. OTA updates allow wireless firmware management.

Future Improvements (Brief)

- Add direct IR raw capture and transmit to support non-NEC devices.

Download

[Download PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/iothings/proiecte/2025src/esp32-ir-hub>



Last update: **2025/05/21 01:03**