

ESP32 Mesh Network for Halloween Prop Control

This project uses a **mesh network of ESP32 microcontrollers** communicating via **ESP-NOW** to coordinate a set of Halloween props. The goal is to create synchronized, responsive effects triggered by motion — perfect for haunted house setups or outdoor displays.

System Overview

The network consists of one **master controller** and multiple **slave controllers**:

- **Master Node:**

Acts as the central coordinator. It's equipped with a motion sensor (such as a PIR sensor) that detects when someone approaches.

When motion is detected, the master broadcasts a **“trigger” message** over ESP-NOW to all slave nodes.

- **Slave Nodes:**

Each slave node controls an individual prop (e.g., fog machine, servo motor, LED lighting, sound module, etc.).

Upon receiving the master's **trigger message**, the slave performs its assigned action — for example, moving a prop, flashing lights, or playing a sound.

Once the action is complete, the slave sends a **“completed” message** back to the master.

Communication Flow

1. **Motion Detection:**

The master's PIR sensor detects movement.

2. **Broadcast Trigger:**

The master sends a “motion detected” message via ESP-NOW to all registered slave devices.

3. **Action Execution:**

Each slave performs its programmed effect or animation.

4. **Completion Feedback:**

After finishing, each slave sends a “done” signal to the master.

5. **Reset or Cooldown:**

Once the master receives all “done” signals (or after a timeout), it resets the system and waits for the next motion event.

Key Features

- **ESP-NOW protocol** allows for **low-latency, Wi-Fi-free** communication — no router or internet required.
- **Scalable design** — easily add or remove props by pairing additional ESP32s.
- **Bidirectional communication** ensures the master knows when all props have completed their sequences.
- **Energy-efficient** — nodes can sleep between triggers.
- **Highly customizable** — each slave can have its own timing, behavior, or randomized effect.

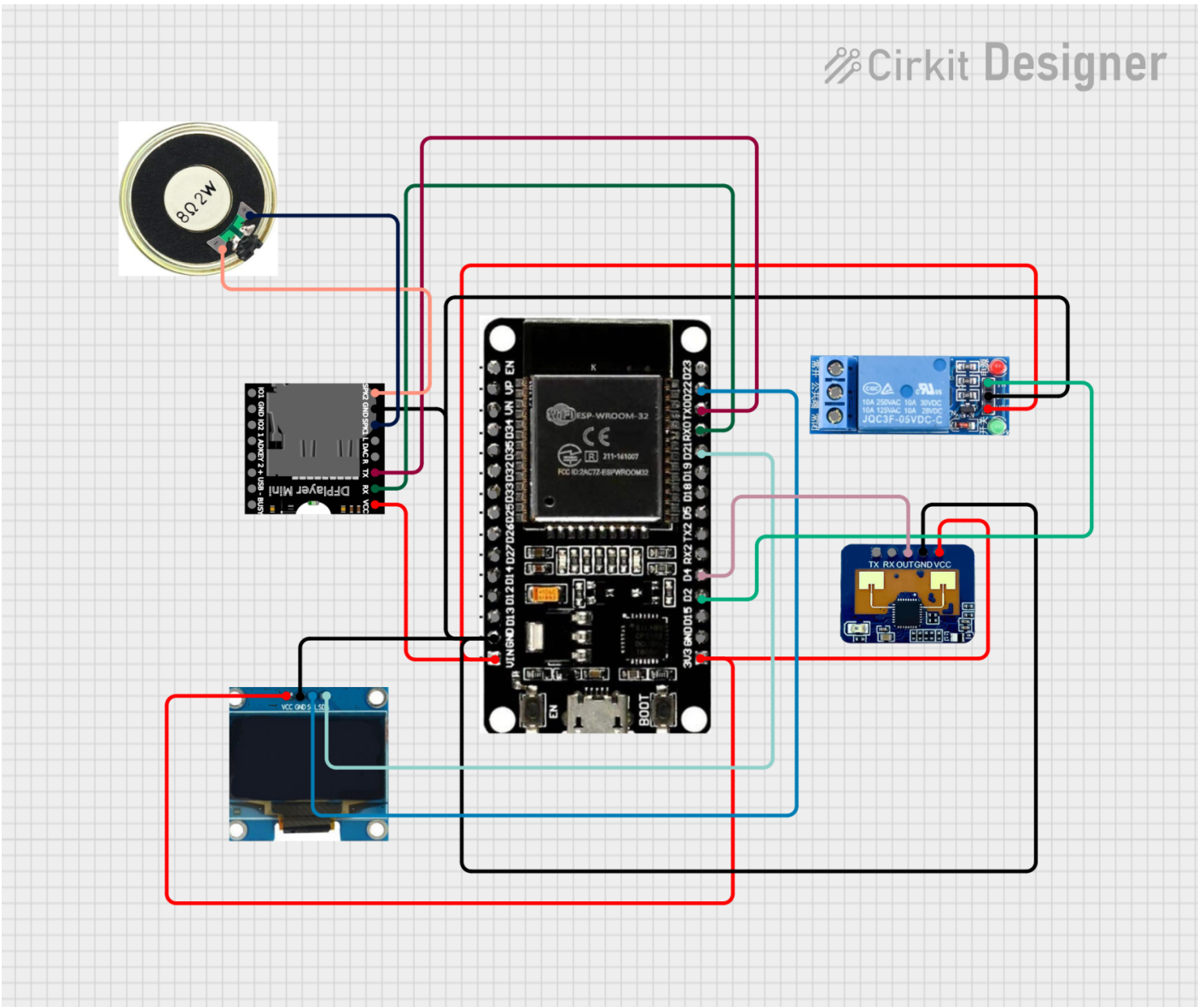
Example Use Case

- Master detects motion → broadcasts “TRIGGER”
- Slave 1 (fog machine): starts fog → sends “DONE” after 10 seconds
- Slave 2 (LED lights): flashes red and orange → sends “DONE” after 5 seconds
- Slave 3 (servo skeleton): waves arm → sends “DONE” after 8 seconds
- Once all “DONE” signals are received, the master resets and waits for the next visitor.

List of mac addresses currently in use:

master (detects motion): FC:B4:67:55:A5:24

sound player: 08:D1:F9:EE:A0:B0



Revision #4

Created 18 October 2025 00:07:26 by peterd

Updated 18 October 2025 22:22:51 by peterd