

Informe del Prototipo V1.0: Sistema de Alarma IoT con Detección de Movimiento y Registro de Eventos

Fecha de Elaboración: 15 de octubre de 2025

1. Síntesis del Prototipo

La versión inicial de este prototipo de Alarma IoT presenta un sistema funcional cuyo principio de operación se fundamenta en la **detección de movimiento a través de un sensor Infrarrojo Pasivo (PIR)**. El sistema ha sido diseñado para identificar posibles intrusiones, implementando una lógica de confirmación temporal para mitigar la incidencia de falsos positivos y activando subsecuentemente una alerta audiovisual. La arquitectura del software se basa en un modelo de ejecución no bloqueante, y cuenta con conectividad Wi-Fi para la **sincronización de la hora a través de NTP (Network Time Protocol)**, permitiendo el registro de eventos con marcas de tiempo precisas (timestamp), garantizando así su escalabilidad futura para la integración con bases de datos y dashboards.

2. Componentes de Hardware Empleados

- **Unidad de Microcontrolador:** ESP32 DevKit
- **Sensor Primario:** Sensor de Movimiento Infrarrojo Pasivo PIR HC-SR-501
- **Actuadores de Alerta:**
 - Diodo Emisor de Luz (LED) de color rojo para indicación de alarma confirmada.
 - Zumbador (Buzzer) pasivo con tres terminales (+, -, S).
- **Componentes Pasivos:**
 - Una resistencia de 220Ω para la limitación de corriente del LED.
- **Infraestructura de Prototipado:**
 - Dos placas de pruebas (protoboard) de diseño modular.
 - Cables de conexión Jumper.

3. Esquema de Interconexiones (Pinout)

Con el fin de asegurar la estabilidad operativa y prevenir conflictos de hardware, se ha realizado una selección estratégica de pines GPIO que carecen de funciones especiales conflictivas.

Sensor PIR (HC-SR-501)

- VCC → **VIN** del ESP32 (Alimentación de 5V)
- GND → **GND** del ESP32 (Referencia a tierra)
- OUT → **GPIO 23** (Terminal de salida de señal de entrada)

LED Rojo (Indicador de Alarma)

- **Terminal Ánodo (+)** → Resistencia 220Ω → **GPIO 22** (Terminal de salida de control)
- **Terminal Cátodo (-)** → **GND** del ESP32 (Referencia a tierra)

Zumbador Pasivo

- **Terminal +** → **VIN** del ESP32 (Alimentación de 5V)
- **Terminal -** → **GND** del ESP32 (Referencia a tierra)
- **Terminal S** → **GPIO 13** (Terminal de salida para generación de tono)

4. Calibración y Parametrización del Sensor

La fiabilidad operativa del sistema está intrínsecamente ligada a la calibración precisa de los parámetros físicos del sensor PIR.

- **Modo de Disparo (Jumper):** Se ha configurado en la posición '**H**' (**Modo Repetible**) para mantener la señal activa mientras persista el movimiento.
- **Sensibilidad (Sx):** El nivel de sensibilidad ha sido ajustado a un valor **medio-bajo**, calibrado específicamente para cubrir el área de prueba designada sin incurrir en activaciones espurias.
- **Tiempo de Retardo (Tx):** Este parámetro se ha configurado a su **valor mínimo**, delegando la gestión de la temporización de la señal al software del microcontrolador para obtener un control más preciso y adaptable.

5. Lógica de Software y Parámetros Fundamentales

El software fue desarrollado con un enfoque en la robustez, la eficiencia y la conectividad.

- **Conectividad y Sincronización Horaria:** Al iniciar, el sistema se conecta a una red Wi-Fi y sincroniza su reloj interno con un servidor NTP. **Todos los eventos de detección son registrados en el monitor serie con una marca de tiempo (timestamp) en formato YYYY-MM-DD HH:MM:SS.**
- **Bucle de Ejecución No Bloqueante:** Se utiliza la función `millis()` para una gestión del tiempo eficiente que previene el bloqueo del procesador.
- **Frecuencia de Muestreo (checkInterval):** Establecido en 100 ms, permitiendo que el sistema verifique el sensor diez veces por segundo.
- **Lógica de Confirmación de Detección (confirmationDelay):** Se ha instituido un retardo de confirmación de 750 ms para validar la detección continua de movimiento antes de activar la alarma, constituyendo la principal estrategia para la supresión de falsos positivos.