Informe Sistema de Detección de Incendios

Proyecto Internet de les coses Gr.A-T (36422)

Fecha: Enero 2024

Elaborado por: Andreu Cantó Belda, Benito Pastor Sánchez y Amparo Gálvez Vilar

Contenido

Introducción

Este proyecto propone un sistema IoT avanzado basado en el hardware PYNQ-Z2, que combina sensores ambientales, procesamiento en tiempo real y notificaciones automatizadas para identificar y reportar posibles incendios de manera eficiente. El sistema propuesto mide la temperatura y la calidad del aire mediante sensores, analiza los datos en tiempo real, y utiliza tecnologías IoT como MQTT y Node-RED para generar notificaciones por voz a través de Alexa y alertas en Discord. Este informe detalla todos los componentes, metodologías y resultados obtenidos en el desarrollo del sistema.

Objetivos del Proyecto

1. Diseñar un sistema IoT basado en PYNQ-Z2 para detectar incendios a través de sensores de temperatura y partículas.
2. Implementar un procesamiento eficiente en tiempo real que permita respuestas rápidas a condiciones críticas.
3. Establecer un sistema de comunicación que notifique automáticamente alertas a usuarios finales a través de múltiples plataformas.
4. Garantizar escalabilidad y adaptabilidad para futuras ampliaciones del sistema.

1. Arquitectura del Sistema

La arquitectura de la aplicación está diseñada para monitorear la temperatura y las partículas del ambiente, y detectar posibles incendios mediante la integración de varios componentes de hardware y software. A continuación, se describe la estructura del sistema:

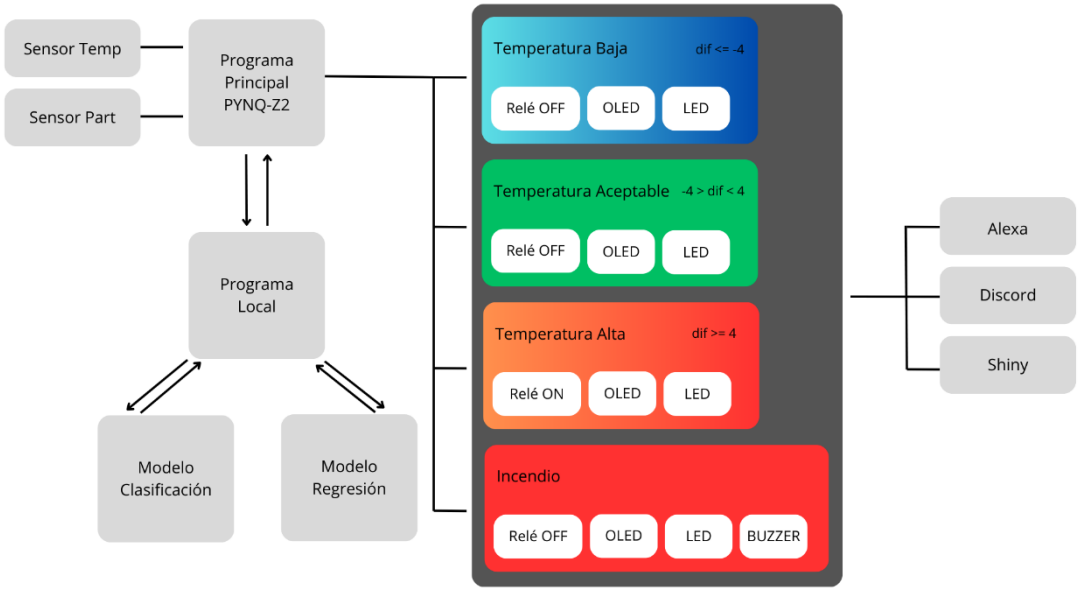

Imagen 1: Esquema de ejecución

1.1. Hardware

El sistema está basado en la PYNQ-Z2, una placa de desarrollo que integra un FPGA con un procesador ARM Cortex-A9. Esta placa se conecta a diversos sensores y actuadores para recopilar datos ambientales y gestionar la respuesta en tiempo real:

- **Sensores:**
    - **Sensor de temperatura**: Mide la temperatura ambiente.
    - **Sensor de partículas**: Mide la concentración de partículas en el aire que pueden indicar la presencia de humo o contaminantes.
- **Actuadores:**
    - **LEDs**: Indicadores visuales que muestran el estado del sistema (normal, ajuste, emergencia).
    - **Pantalla OLED**: Muestra los valores de temperatura y partículas, además de la alerta actual (normal, emergencia, ajuste).
    - **Relé de control**: Controla el encendido y apagado de un ventilador (de 3 o 5V) para regular la temperatura en caso de detectar valores anómalos.
- **Switches**: Actuador manual para detener el ciclo de medición.

1.2. Software

El software del sistema se ejecuta en la PYNQ-Z2 y se divide en varios módulos que gestionan la recopilación de datos, el procesamiento de la información y la notificación a los usuarios:

- **Node-RED**: Utilizado para la integración y gestión de los flujos de trabajo. Recibe los datos de la PYNQ-Z2 a través de MQTT y los utiliza para generar alertas y notificaciones.
- **MQTT**: Protocolo de comunicación ligero que permite a la PYNQ-Z2 intercambiar datos con Node-RED en tiempo real.
- **Alexa**: Se utiliza para enviar mensajes de voz a los dispositivos Alexa en el entorno, notificando a los usuarios sobre situaciones críticas.
- **Discord**: Plataforma de mensajería que permite enviar alertas detalladas a través de un webhook.
- **Shiny**: Permite visualizar los datos de los sensores, como la temperatura y las partículas en el aire, en tiempo real.

1.3. Conexión y Comunicaciones

El sistema utiliza varias tecnologías de comunicación:

- **MQTT**: Para enviar datos desde la PYNQ-Z2 hacia Node-RED y local. Este protocolo es eficiente y permite una comunicación rápida y confiable entre dispositivos.
- **API de Alexa y Webhooks de Discord**: Para enviar alertas a través de los asistentes virtuales (Alexa) y plataformas de mensajería (Discord), de forma que los usuarios puedan recibir notificaciones instantáneas sobre los eventos críticos.

2. Descripción Detallada de Cada Componente

2.1 Sensores

Los sensores seleccionados son fundamentales para garantizar mediciones precisas. Cada uno de ellos se conecta directamente a los pines GPIO de la PYNQ-Z2 y opera bajo los siguientes principios:

- **Sensor de temperatura:** Este sensor mide la temperatura ambiente.
- **Sensor de partículas:** Detecta las partículas suspendidas en el aire.

2.2 Actuadores y Dispositivos de Salida en el Sistema

En el sistema, los actuadores y dispositivos de salida desempeñan un papel clave en la interacción con el entorno. Basándose en las lecturas de sensores de temperatura y partículas, y los cálculos realizados por el modelo de predicción, se realizan diferentes acciones mediante:

1. **LEDs RGB**: Representan visualmente el estado del sistema y la diferencia entre la temperatura medida y la ideal.
2. **Relé**: Activa o desactiva un ventilador de bajo voltaje.
3. **Pantalla OLED**: Proporciona una interfaz visual con información detallada sobre el estado del sistema.
4. **Alarma y luces**: Generan señales de advertencia en caso de detección de incendio.
5. **Alexa**: Comunica alertas por voz de manera proactiva.

2.2.1 LEDs RGB

Los LEDs RGB sirven como indicadores visuales en tiempo real. Dependiendo de la diferencia entre la temperatura medida y la ideal (dif), se ajustan los colores mostrados:

- **Verde**: Cuando la temperatura está dentro del rango ideal (±4 °C), indicando normalidad.
- **Rojo**: Cuando la temperatura es significativamente alta (por encima de +4 °C), sugiriendo un ambiente peligroso.
- **Azul**: Cuando la temperatura es significativamente baja (por debajo de -4 °C), indicando la necesidad de aumentar la calefacción o tomar precauciones.

Esta lógica ajusta dinámicamente el color y actualiza todos los LEDs para reflejar el estado del sistema.

2.2.2 Relé

El relé es un actuador clave para controlar dispositivos externos, en nuestro caso un ventilador de bajo voltaje. Se activa o desactiva según el estado del sistema:

- **Temperatura alta (dif >= 4)**: El relé se activa (rele.on()) para encender el sistema de refrigeración.
- **Temperatura dentro del rango normal**: El relé permanece apagado (rele.off()).
- **Incendio detectado**: El relé se apaga como medida de seguridad para evitar el uso de equipos eléctricos.

2.2.3 Pantalla OLED

La pantalla OLED es un componente crítico para mostrar información detallada en tiempo real. Se utiliza para mantener a los usuarios informados sobre el estado actual del sistema, presentando datos como:

- **Temperatura medida**.
- **Diferencia entre la temperatura medida y la ideal**.
- **Temperatura ideal calculada**.
- **Estado del sistema** (normal, ajuste, emergencia).

El método actualizar_pantalla() permite actualizar la pantalla OLED con diferentes tipos de mensajes:

- **Estado normal**: Temperatura en el rango ideal.
- **Ajuste**: Temperatura fuera del rango ideal, pero no crítico.
- **Emergencia**: Incendio detectado.

2.2.4 Alarma y Luces

Cuando el sistema detecta un incendio (if incendio:), se activa un procedimiento de emergencia que incluye:

Encendido de luces y alarma sonora mediante la función sonido_luces_alarma():

- Las luces parpadean para captar la atención de las personas cercanas.
- La alarma emite un sonido continuo durante 15 segundos para alertar sobre el peligro.
- Se emplea un retraso (delay_luces=10) para intermitencia en las luces.

Mensajes de voz por Alexa: Alexa emitirá un mensaje de alerta.

2.2.5 Alexa: Comunicación por Voz

Alexa se utiliza para proporcionar mensajes de alerta personalizados basados en el estado del sistema:

- Temperatura baja: "¡Atención!, temperatura muy baja detectada. Abríguese."
- Temperatura alta: "¡Atención!, temperatura elevada, encendiendo sistema de refrigeración."
- Incendio detectado:"¡Atención!, Incendio detectado. ¡Evacúe el área inmediatamente!"

Alexa complementa las acciones de los actuadores proporcionando una capa de notificación adicional, que es útil especialmente en situaciones de emergencia.

3. Comunicación y Notificaciones

3.1 MQTT (Message Queuing Telemetry Transport)

En este proyecto, el protocolo MQTT es utilizado para establecer una comunicación bidireccional entre la PYNQ-Z2 y el servidor Node-RED, que se encarga de procesar estos datos y generar notificaciones. Los mensajes se publican en temas (topics) específicos que se suscriben en Node-RED para procesar y distribuir las alertas.

Temas MQTT utilizados:

- **PYNQ-Z2/alexa:** Este tema es utilizado para transmitir los datos procesados y las alertas de notificación desde la PYNQ-Z2 a Node-RED. Node-RED se encarga de generar las notificaciones, las cuales son enviadas a sistemas como Alexa para que se activen las alertas de voz y Discord para enviar notificaciones en tiempo real.
- **Datos_sensores:** Este es el tema a través del cual la PYNQ-Z2 publica los datos obtenidos por los sensores (temperatura y concentración de partículas). Estos datos son escuchados por el servidor local, que se encarga de procesarlos utilizando modelos de machine learning para detectar situaciones de emergencia, como incendios o temperatura ideal.
- **Datos_modelos**: Este tema se utiliza para enviar los resultados del procesamiento realizado en el servidor local. Tras aplicar los modelos de machine learning sobre los datos obtenidos, el servidor local envía las predicciones (como la probabilidad de incendio o ajustes de temperatura) de vuelta a la PYNQ-Z2 a través del topic Datos_modelos. La PYNQ-Z2 escucha este tema para recibir las predicciones y ajustar las acciones del sistema según sea necesario.

3.2 Node-RED: Integración y Generación de Notificaciones

El flujo de trabajo en Node-RED se estructura mediante nodos interconectados, cada uno con un rol específico para garantizar que los datos se reciban correctamente, se procesen y se envíen como notificaciones. El flujo general incluye nodos para la integración de la PYNQ-Z2, la interacción con Alexa y Discord, así como la depuración de mensajes para asegurar que cada parte funcione como se espera.

3.2.1 Flujo de Node-RED:

En la captura siguiente captura, se verá la estructura de flujo de Node-RED diseñada para manejar las notificaciones y comunicaciones. A continuación, se explica cómo está estructurado el flujo de trabajo:

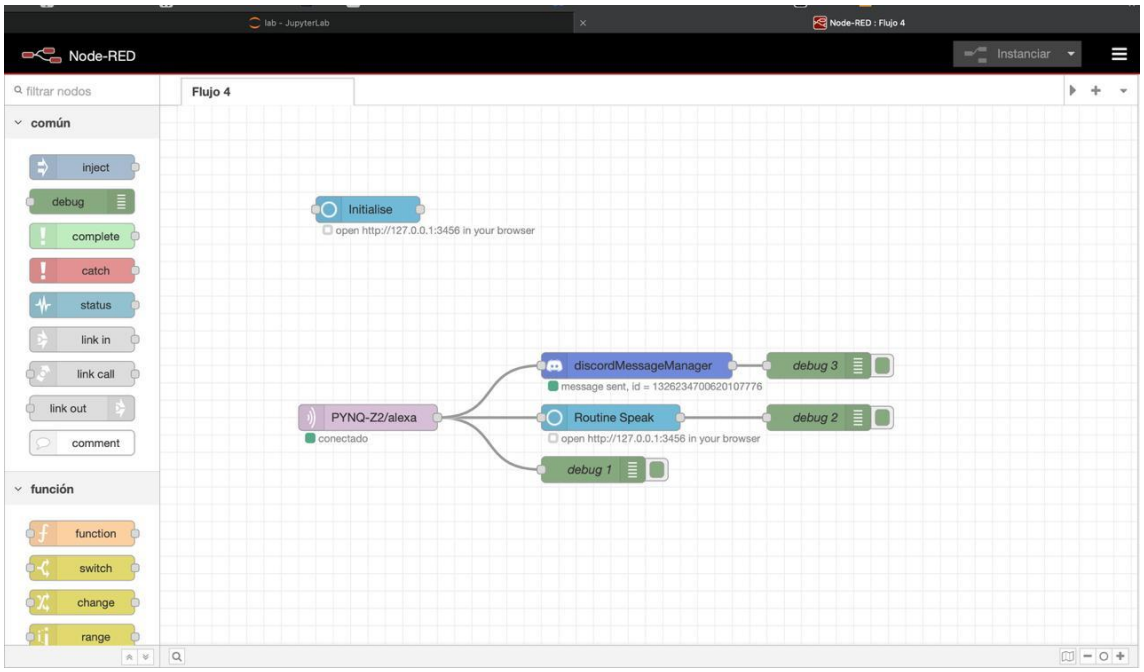

Imagen 2: Diagrama de flujo en Node-RED

En Node-RED, todo lo que llega desde la PYNQ-Z2 ya está procesado y llega con el tema "PYNQ-Z2/alexa". Este tema contiene la información necesaria para activar las notificaciones y alertas en el sistema. En Node-RED no se realiza procesamiento adicional de los datos, ya que estos ya han sido analizados en el servidor local. El flujo se limita a recibir los datos y gestionar las alertas, enviándolas a Alexa y Discord.

Nodo MQTT: Recibe los datos MQTT que se publican en el topic “PYNQ-Z2/alexa”.

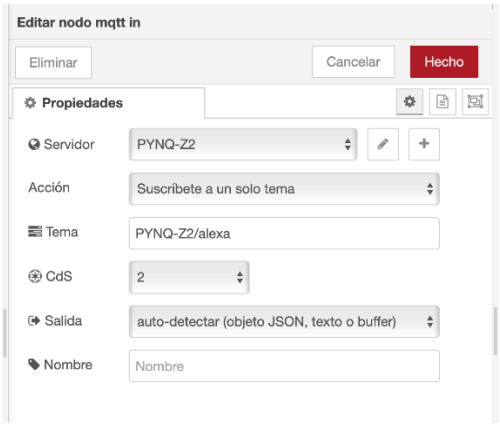

Imagen 2: Configuración MQTT

Nodo debug 1: Depura los mensajes que vienen del nodo PYNQ-Z2/alexa, permitiendo ver los datos recibidos antes de que sean enviados a Alexa para asegurarse de que la información es correcta.

Nodo discordMessageManager: Recibe los datos procesados y genera un mensaje de alerta estructurado para enviarlo a un canal de Discord a través de un webhook. Enviando una serie de alertas que se activan según las condiciones de temperatura y posibles riesgos detectados por el sistema.

Nodo debug 3: Depura los mensajes antes de ser enviados a Discord, permitiendo verificar que el mensaje tiene el formato correcto y asegurándose de que la información es precisa.

Nodo Routine Speak: Ejecuta las alertas de voz en Alexa. Si se detecta un incendio, envía un mensaje predefinido, como "¡Alerta! Se ha detectado un incendio." También puede emitir otros mensajes informativos sobre la temperatura o partículas.

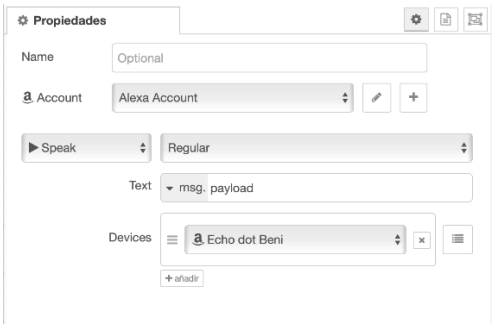

Imagen 3: Configuración de Alexa

Nodo debug 2: Depura el mensaje que se enviará a Alexa, asegurando que está correctamente formateado antes de ser reproducido en el dispositivo.

Este flujo de nodos facilita la integración de la PYNQ-Z2 con Alexa y Discord para notificaciones en tiempo real, garantizando que los usuarios reciban alertas sobre el estado del sistema.

3.3 Mensajes Generados:

Alexa:

Cuando el sistema detecta una situación de emergencia, como un incendio, se envía un mensaje de voz a los dispositivos Alexa que estén configurados para recibir estas alertas. Los usuarios escuchan un mensaje como el siguiente:

> "¡Alerta! Se ha detectado un incendio. Revise la temperatura y calidad del aire."

Este mensaje ayuda a alertar a los usuarios en tiempo real, permitiendo una respuesta rápida y eficiente.

Discord:

De manera simultánea, Node-RED envía un mensaje al servidor de Discord a través de un webhook. El mensaje enviado es una serie de alertas que se activan según las condiciones de temperatura y posibles riesgos detectados por el sistema.

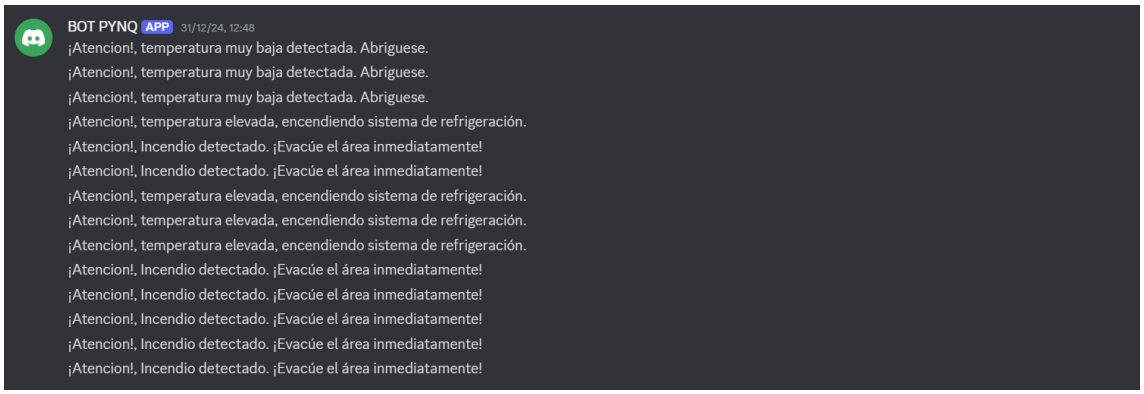

Imagen 4: Ejemplo mensajes en Discord

El flujo de mensajes refleja diferentes situaciones críticas que pueden ocurrir durante la supervisión del entorno:

1. **Temperatura Baja**: Si el sistema detecta que la temperatura es muy baja, se genera el mensaje: *¡Atención!, temperatura muy baja detectada. Abríguese.*
2. **Temperatura Alta**: Si la temperatura supera el umbral de seguridad, el sistema avisa con el mensaje: *¡Atención!, temperatura elevada, encendiendo sistema de refrigeración.*
3. **Incendio Detectado**: En caso de que el sistema detecte un incendio a través de los sensores de temperatura o partículas, se emite una alerta urgente con el siguiente mensaje: *¡Atención!, Incendio detectado. ¡Evacúe el área inmediatamente!*

Estos mensajes se repiten dependiendo de las condiciones que se detectan en el sistema. La repetición garantiza que los usuarios reciban alertas constantes hasta que el problema sea resuelto o el sistema cambie de estado.

El envío repetido de los mensajes asegura que no se pase por alto ninguna alerta crítica. Esto permite una respuesta rápida a situaciones de emergencia, mejorando la seguridad en el entorno monitoreado.

4. Shiny: Interfaz de Usuario para Visualización

Shiny es una herramienta de R utilizada para crear aplicaciones web interactivas. En este proyecto, se ha integrado Shiny para ofrecer una visualización en tiempo real de los datos obtenidos por los sensores de temperatura y partículas, así como los resultados de las predicciones y clasificaciones generadas por los modelos.

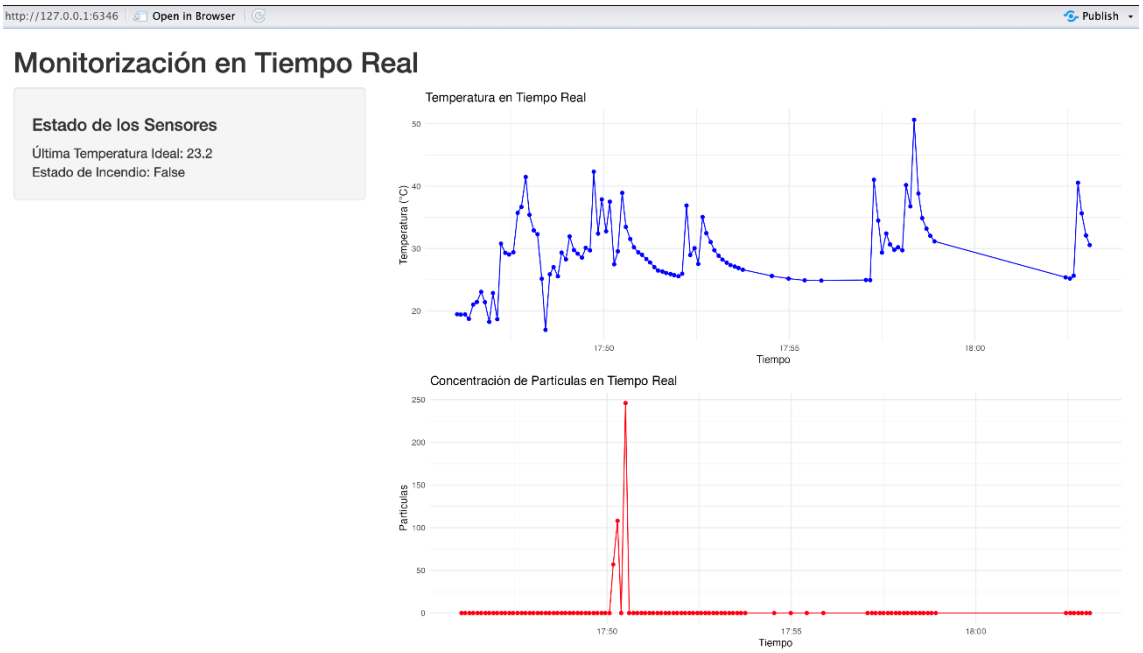

Imagen 5: Ejemplo salida de Shiny

Características Principales

- **Visualización en Tiempo Real:** Muestra los datos de temperatura y partículas en gráficos dinámicos.
- **Alertas:** Informa visualmente sobre temperaturas extremas o concentración elevada de partículas mediante las gráficas.
- **Generación de CSV**: Cuando los modelos realizan la predicción o clasificación correspondiente, la salida de estos se escribe en un archivo CSV. Este archivo es utilizado para alimentar los gráficos y visualizaciones en la aplicación Shiny, permitiendo mostrar los valores procesados, como las predicciones de temperatura y las clasificaciones de incendio, junto con los datos de los sensores.

5. Flujo de Decisiones y Procesamiento y Análisis

El sistema sigue un flujo lógico para procesar los datos de los sensores y tomar decisiones basadas en los valores obtenidos. A continuación, se detalla el flujo de decisiones y cómo se procesan los datos.

5.1. Recopilación de Datos

Cada ciclo del sistema comienza con la recopilación de datos de los sensores de temperatura y partículas. Estos datos son procesados y ajustados en la PYNQ-Z2:

- **Temperatura**: La temperatura medida por el sensor es ajustada para compensar posibles desviaciones causadas por el tipo de alimentación del sensor.

- **Partículas**: La concentración de partículas se lee del sensor y se envía junto con la temperatura para su procesamiento.

5.2. Procesamiento de Datos (Modelo de Predicción)

Una vez que los datos son recogidos, se envían a un modelo de predicción en local, que realiza un análisis de la situación en base a los datos recibidos. Este modelo evalúa tres variables:

- **Tideal**: Temperatura ideal calculada por el modelo.
- **Incendio**: Indicador binario que indica si se ha detectado un incendio o no.
- **Iteración**: El número de iteración del ciclo de lectura y análisis.

El procesamiento de estos valores es crucial para determinar la respuesta del sistema ante un posible incendio.

5.3. Decisiones Basadas en los Datos

Basado en los valores de temperatura y partículas y la salida del modelo, el sistema toma decisiones de actuación que afectan a los actuadores:

Estado Normal:

- Si la temperatura se encuentra dentro de un rango aceptable (diferencia entre la temperatura medida y la temperatura ideal es pequeña), el sistema se mantiene en modo normal.
- Los LEDs se mantienen apagados o en un color normal.
- La pantalla OLED muestra los valores de temperatura y partículas en un formato normal.
- El relé (ventilador) se apaga.

Temperatura Baja:

- Si la temperatura está por debajo de un umbral específico (diferencia de más de 4 grados con la temperatura ideal), el sistema alerta sobre una temperatura baja.
- Los LEDs y la pantalla OLED cambian para reflejar el estado de "baja temperatura".
- El relé se apaga (sin acción de enfriamiento).
- Se envía un mensaje a Alexa y Discord para alertar sobre la baja temperatura.

Temperatura Alta:

- Si la temperatura excede un umbral (diferencia de más de 4 grados con la temperatura ideal), el sistema activa el relé para encender el ventilador.
- Los LEDs y la pantalla OLED cambian para reflejar el estado de "ajuste de temperatura".
- Se envía un mensaje a Alexa y Discord alertando sobre la temperatura elevada.

Incendio Detectado:

- Si el modelo predice que un incendio está ocurriendo (se detecta un valor de temperatura y/o partículas que superan los umbrales críticos), el sistema entra en emergencia.

- El relé se apaga, y los LEDs se configuran para reflejar el estado de emergencia.
- La pantalla OLED muestra el estado de emergencia.
- Se envía un mensaje de alerta urgente a Alexa y Discord, y se activa la alarma y luces del sistema para alertar a los usuarios.

5.4. Comunicación de Alertas

Cuando el sistema detecta un estado de alerta, se generan notificaciones en tiempo real:

- **Alexa**: El mensaje de alerta es transmitido por voz, permitiendo que los usuarios sean informados de manera rápida sobre las condiciones peligrosas.
- **Discord**: Los detalles del evento se envían a un canal de Discord, incluyendo la temperatura registrada, la concentración de partículas y la hora del evento, para que los usuarios puedan monitorear la situación en tiempo real.

6. Conclusión

El sistema diseñado y desarrollado proporciona una solución eficiente y robusta para la detección de incendios a través de sensores de temperatura y partículas, integrando múltiples tecnologías y protocolos de comunicación como MQTT, Node-RED, y Shiny. Mediante el uso del PYNQ-Z2 y la conexión con plataformas como Alexa y Discord, el sistema no solo monitoriza el ambiente en tiempo real, sino que también proporciona alertas inmediatas y detalladas a los usuarios, mejorando la capacidad de respuesta ante situaciones de emergencia.

El flujo de decisiones automatizado, que ajusta los actuadores según las lecturas de los sensores, demuestra la capacidad del sistema para actuar de manera autónoma y eficiente. Además, la interfaz visual en Shiny ofrece una representación clara de los datos, permitiendo un monitoreo continuo y facilitando la toma de decisiones.

Este proyecto resalta la importancia de la integración de tecnologías IoT, la comunicación en tiempo real y la interacción con dispositivos inteligentes para mejorar la seguridad y la gestión de emergencias. Sin embargo, el sistema podría beneficiarse de mejoras en la precisión de los sensores, la optimización del procesamiento de datos y la ampliación de su alcance a otras plataformas de notificación.