GRUPODATCO

Documentación del proyecto de práctica

Santiago, febrero del 2022

Índice

Proyecto

El proyecto de práctica fue investigar qué era MQTT e implementarlo, con el objetivo de enviar datos desde un ESP32 a un servidor y, a la vez, recibir datos en un ESP32 desde el servidor. Para lograr esto, se utilizó el broker EMQ X y se instaló en un servidor de Datco. Luego, se programó el ESP32 conectado a un MAXIM6675, el que a su vez estaba conectado a una termocupla tipo K para enviar datos de temperaturas al servidor previamente instalado. También, se programó el ESP32 conectado a un relé para recibir datos desde el servidor y activar o desactivar el switch del relé.

Una vez se logró este objetivo, se creó una plataforma web para monitorear las temperaturas enviadas por el ESP32 y para poder cambiar el estado del relé con un botón. Posteriormente, se programó un script en Python para subir cada publicación a una base de datos, con el fin de utilizar estos datos para crear gráficos que muestren la tendencia y así, poder hacer estudios y analizar el comportamiento de las temperaturas.

Herramientas utilizadas en el proyecto

Lenguajes de programación:

- C
- HTML5
- CSS
- JS
- PHP
- Python

Dispositivos utilizados:

- ESP32
- MAXIM6675
- Termocupla tipo K
- Relé
- Servidor de Datco en dirección: 190.110.108.59

Sistema de administración de bases de datos:

- MySQL

Requisitos técnicos del servidor utilizado:

- CPU: Procesador de 64 bits a 2GHz (2 núcleos)
- RAM: 2 GB
- Almacenamiento: 64 GB
- Red: Adaptador Ethernet de 1 gigabit/s
- Sistema operativo: Windows Server 2022
- Puertos habilitados: 1883, 8083, 18083

MQTT

¿Qué es?

MQTT es el estándar para OASIS, para la conectividad del Internet de las Cosas (IoT). Es un protocolo de mensajería de tipo publicador/suscriptor extremadamente simple y ligero diseñado para dispositivos restringidos y de poca banda ancha, alta latencia o que utilicen redes poco confiables.

Los principios de este diseño se utilizan para minimizar el ancho de banda de la red y los requisitos de recurso de los dispositivos mientras que se intenta garantizar la confiabilidad y cierto grado de garantía de entrega. Estos principios hacen que el protocolo sea ideal para el mundo de los dispositivos conectados del "Internet de las cosas", y para las aplicaciones móviles donde el ancho de banda y la energía de la batería son escasos.

QoS

MQTT dispone de un mecanismo llamado Quality of Service (en español, calidad del servicio), que se entiende como la forma de gestionar la robustez del envío de mensajes al cliente ante fallos. Existen 3 niveles de QoS:

- QoS 0 (at most one): El mensaje se envía una única vez. En caso de fallo no se vuelve a enviar, por lo que puede que alguno de los mensajes enviados no se entregue.
- QoS 1 (at least one): El mensaje se envía hasta que se garantiza la entrega. En caso de fallo, el suscriptor puede recibir mensajes duplicados.
- QoS 2 (exactly one): Se garantiza que cada mensaje se entrega al suscriptor, y únicamente una vez.

Broker

Un Broker es un servidor que enruta los mensajes publicados a los suscriptores.

Modelo publicador/suscriptor

El modelo publicador/suscriptor es un patrón de mensajería en el que los remitentes de mensajes, llamados publicadores, no programan los mensajes para que se envíen directamente a receptores específicos (llamados suscriptores), sino que clasifican los mensajes publicados en clases sin saber qué suscriptores puede haber. Del mismo modo, los suscriptores expresan interés en una o más clases y solo reciben mensajes que son de su interés, sin saber qué publicadores hay.

¿Cómo funciona MQTT?

Como se mencionó anteriormente MQTT implementa el modelo publicador/suscriptor, de la siguiente manera:

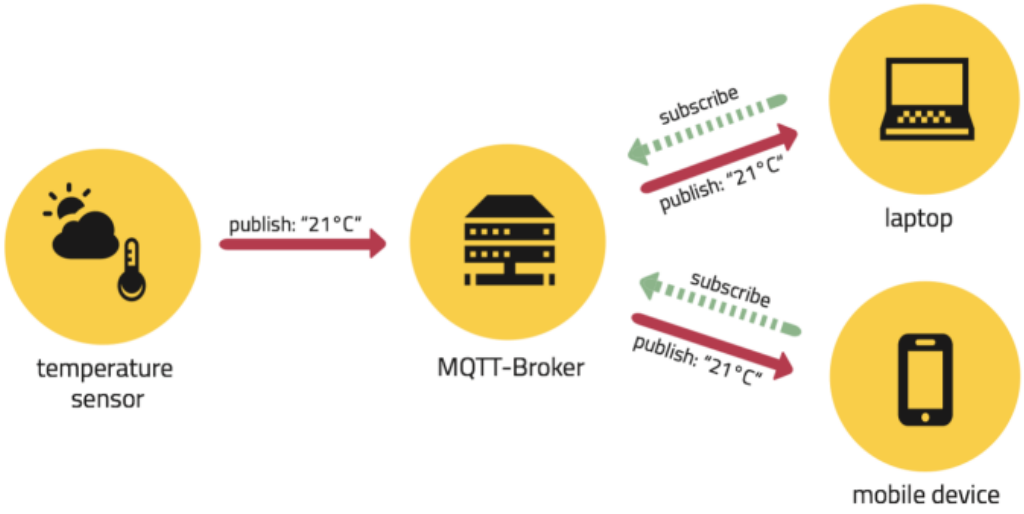

Imagen 1

La imagen 1 es un diagrama de MQTT que implementa un modelo publicador/suscriptor en donde el sensor de temperatura es el publicador y envía los datos de temperatura al MQTT-Broker, al cual están suscritos el notebook y el celular para recibir estos datos de temperatura.

Topics

Dentro del broker, los mensajes se publican en Topics. Cabe destacar que no es necesario configurar ninguno de estos Topics, sólo publicar en ellos y se representan por una dirección, a la que un suscriptor se va a suscribir para recibir todos los mensajes que un publicador envíe a este Topic. Por ejemplo, si a un suscriptor le interesara saber la temperatura de la oficina 1 de Datco en Chile, tendría que suscribirse al siguiente topic: “Datco/Chile/Oficina_1/Temperatura”.

Wildcards

Dentro de los topics, existen las llamadas “Wildcards”, los que son como comodines para suscribirse a un topic.

Por ejemplo, si quisiera suscribirse a la temperatura de todas las oficinas de Datco en Chile, en lugar de suscribirme una por una a todas las oficinas, simplemente tendría que suscribirme a lo siguiente: “Datco/Chile/+/Temperatura”.

Ahora bien, si quisiera suscribirme a todos los datos que lleguen a la oficina 1 de Datco en Chile (pensando que estamos monitoreando más de una cosa, como temperatura, humedad, presión, etc), tendría que suscribirme a lo siguiente: “Datco/Chile/Oficina_1/#”.

Broker utilizado

EMQ X: Es un proveedor de software de infraestructura de datos de IoT de código abierto en la era 5G. EMQ ofrece el broker de mensajes MQTT de código abierto líder en el mundo y la base de datos de procesamiento de flujo, proporciona una solución integral para el movimiento de datos de IoT en tiempo real, el procesamiento de flujo y el análisis de datos. Tiene una escalabilidad masiva y muy buena, pues es posible conectar decenas de millones de dispositivos IoT a través de un clúster EMQ X de manera eficiente.

Además de lo anteriormente mencionado, EMQ X posee una dashboard de administración propia, lo cual resulta muy útil para administrar, monitorear, manejar y probar el broker de manera remota.

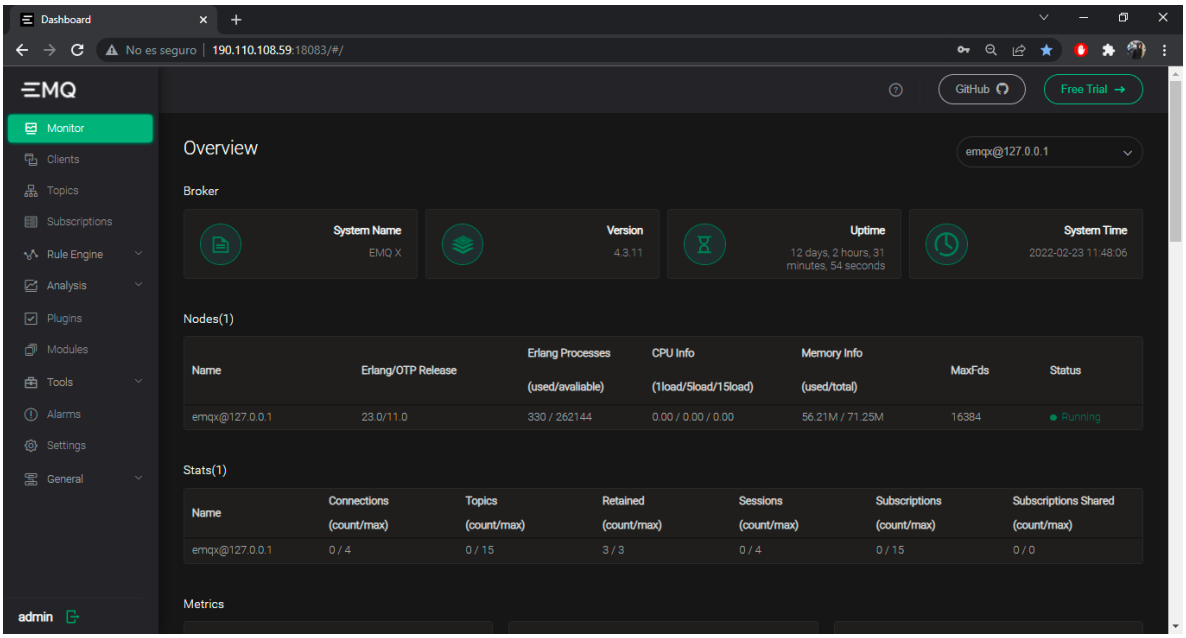

Imagen 2