

IoT-Wifi-MQTT

- [Que es IoT](#)
- [MQTT](#)
- [HiveMQ crear credenciales](#)
- [En Arduinoblocks](#)
- [En cliente: Las APPs posibles](#)
- [Actividad 17 APP IoT MQTT Panel](#)

Que es IoT

El **Internet de las cosas** (Internet of Thing IoT) describe objetos físicos —o grupos de estos— con sensores, capacidad de procesamiento, software y otras tecnologías que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación. El Internet de las cosas se ha considerado un término erróneo porque los dispositivos no necesitan estar conectados a la Internet pública. Sólo necesitan estar conectadas a una red y ser direccionables individualmente

[Fuente Wikipedia IoT Internet de las cosas CC-BY-SA](#)



[De Drawed by Wilgengebroed on FlickrTranslated by Prades97 CC BY-SA 3.0](#)

Estamos hablando de dispositivos que se conectan a internet de forma desatendida, por vía hardware (o mejor dicho firmware) a diferencia de un ordenador, tablet o móvil, donde tienes que configurar por software el dispositivo y hay un diálogo entre usuario y dispositivo sobre el uso de Internet (el software solicita tal página web, tales datos etc por voluntad del usuario o por diálogo con el usuario) Aquí los dispositivos están ya configurados de los datos que se comunican. Es decir "conectar y olvidar".

Piensa en la diferencia entre un enchufe inteligente y un ordenador, el primero es lo que se considera dentro de IoT

Las formas "desatendidas" son un avance en la sociedad pero también puede generar problemas muy serios a nivel mundial, [ver el caso Mirai](#)

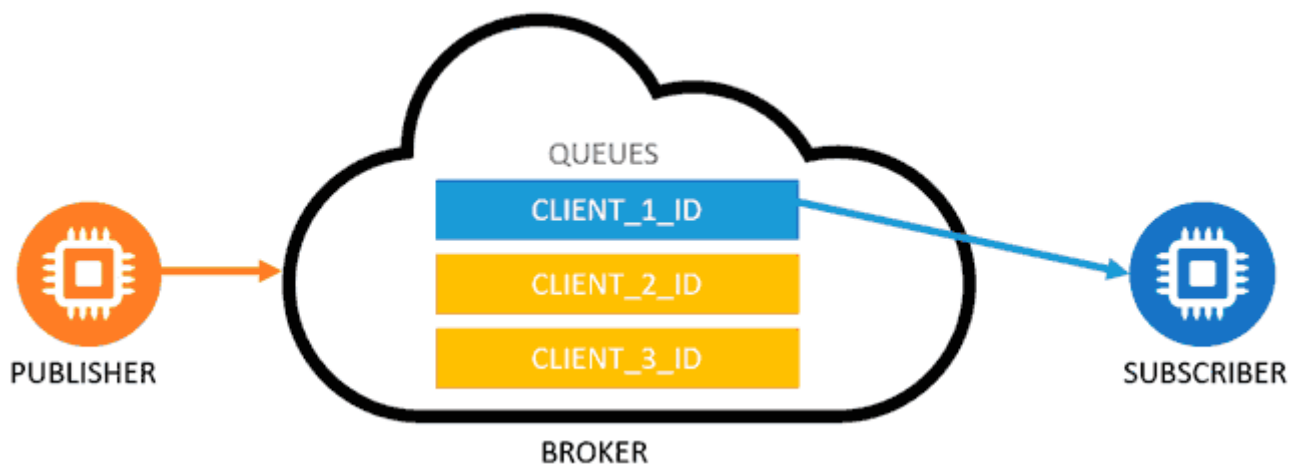
Las cosas claras. ¿asíncrono o síncrono?

Hay muchas herramientas IoT

- **Blynk:** lo que nos gusta de esta herramienta es que es casi "instantánea" o "síncrona". Esto es imprescindible con ciertos robots como el **Rover Marciano con Arduino**. Necesitamos que "gire" para evitar un obstáculo, no podemos esperar !!! Veremos con **BLYNK** un protocolo que entre el dispositivo electrónico (nuestro robot) y nosotros (en ordenador, en una APP en el móvil) la comunicación es instantánea, gracias a un servidor que hará de intermedio, que puede ser local (BLYNK LEGACY) o en Internet (BLYNK IoT).
 - **Blynk legacy** es la que se va a trabajar en
 - [Rover Marciano con Arduino](#)
 - [Arduinoblocks en el aula](#)
 - [ESP32 en el aula](#)
 - **Blynk IoT** es la que se va a trabajar con
 - [En ESP32 en el aula](#)
- **MQTT** El emisor envía datos, se almacenan en un servidor, y cuando puede, lo vuelca al cliente. Cliente y emisor pueden ser el dispositivo electrónico y nosotros o viceversa. Veremos que esto es lo que hace el protocolo **MQTT** y está tremendamente extendido por lo barato y fácil que es. Hace que los servidores no estén tan ocupados, por lo tanto hay varios proveedores que ofrecen este servicio gratuitamente. Hay robots como los que tienen la placa **TDR STEAM IMAGINA** que envía datos de temperatura, humedad, .. y pueden recibir datos pero no precisan de esta exigencia instantánea como un rover.
 - [ESP32 EN EL AULA](#)
- **Arduino cloud IoT**
 - [Arduino Alvik](#)
- **Cyberpi y mBot2**
 - [IoT con Cyberpi](#)

MQTT

MQTT es un protocolo dentro de los protocolos entre máquina a máquina M2M. MQ significa Message Queue, son mensajes que se publican y otros se suscriben. Llegan a un servidor que también se llama **Broker** o Router, los va almacenando en una cola y se eliminan cuando son entregados. Un ejemplo fácil de entender son los mensajes SMS, que mientras no estas conectado, tus mensajes están en un servidor, y cuando te conectas, se descargan tus mensajes en tu móvil. Las últimas siglas TT significa Telemetry Transport



Fuente Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

“ Los clientes inician una conexión TCP/IP con el broker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza. Por defecto, MQTT emplea el puerto 1883 y el 8883 cuando funciona sobre TLS.

Fuente Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

TCP/IP son los protocolos de comunicación en Internet. TLS Transport Layer Security es un protocolo que se añade para asegurar la privacidad y seguridad de los datos. Por eso hay que enviar un usuario y una contraseña para conseguir esta seguridad.

En los mensajes, se añade un texto que haga de filtro, que se llama **topic**, este topic permite al broker filtrar de todos los mensajes que se reciben a qué clientes hay que entregar estos mensajes. Estos topic suelen estar jerarquizados, por ejemplo Casa/Cocina/Temperatura y un cliente puede estar suscrito a Casa/+Temperatura por lo que recibiría los datos de Casa/Cocina/Temperatura pero también los de Casa/Salon/Temperatura, etc... Si quieres saber más de los tópic te recomendamos la página

<https://www.luisllamas.es/que-son-y-como-usar-los-topics-en-mqtt-correctamente/>

El MQTT se ha hecho muy popular en los dispositivos electrónicos, porque:

- Es **libre**, al principio en 1999 era un protocolo propietario y se usaba para la industria petrolera, pero desde el 2010 se liberó y su consecuente popularidad que se ha convertido en un estándar de comunicación electrónica.
- Es **sencillo**, esto es importante para no pedir mucho ancho de banda
- Consume **poca potencia**, esto es importante pues la mayoría están conectados todo el tiempo.
- Un protocolo con una capa de **seguridad**, robusta y largamente testada.

Para saber más te recomendamos la página de Luis Llamas <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

Y el broker ¿de donde?

Puedes montarte tu propio broker en un ordenador, pues es código abierto, por ejemplo instalar Mosquitto <https://mosquitto.org/> en una raspberry.

O bien usar brokers MQTT gratuitos y públicos en Internet, con sus limitaciones de anchos de banda o número de conexiones... como HIVEMQ Info: <http://www.mqtt-dashboard.com/> con la dirección Broker.hivemq.com o también el propio Mosquito <https://mosquitto.org/> con la dirección test.mosquitto.org

Tenemos que tener en cuenta que los servidores MQTT públicos gratuitos no son seguros, cualquiera puede suscribirse a nuestros mensajes y publicar en ellos por lo tanto enviar órdenes. Evitar usar estos brokers en sistemas que no sean educativos y de pruebas.

El protocolo MQTT también se usa en Domótica, por ejemplo [aquí en el curso de Raspberry apartado domótica](#)

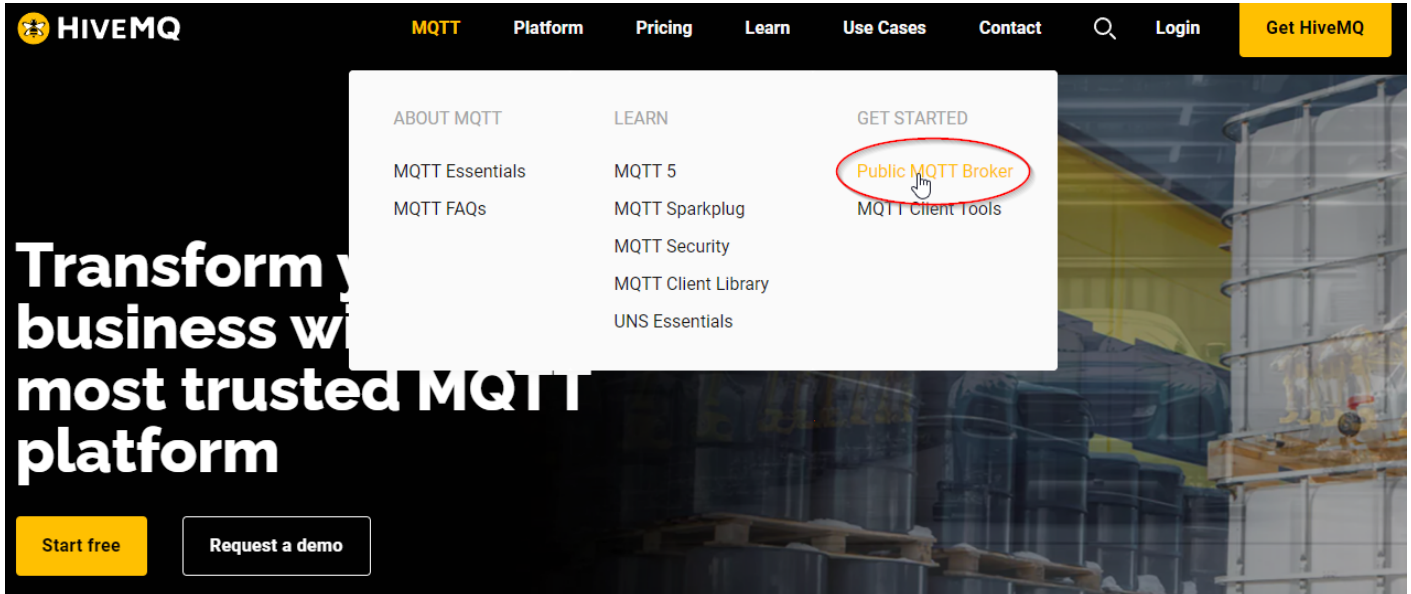
<https://www.youtube.com/embed/YPS5xI6Bx3I>

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



HiveMQ crear credenciales

Entramos en <https://www.hivemq.com/> y nos abrimos una nueva cuenta en Public MQTT Broker



Concretamente en la cuenta gratuita que se puede ver en <https://www.hivemq.com/pricing/>

Pricing

Self-Service

HiveMQ Cloud

Enterprise

Self-Managed and HiveMQ Cloud

HiveMQ Cloud Self-Service

Develop, test, deploy, and scale production IoT use cases without a large investment

Serverless

Free

Get Started

No credit card required

Recommended

Starter

Starting from \$0.34/hour

+ \$0.80/million messages

Get Started FREE

15 day trial - no credit card required

Una vez dentro tenemos que abrir unas credenciales para que nuestra placa pueda usar este broker, en Cluster-Manager cluster y luego Access Management, creamos un **usuario y contraseña** que luego usaremos en Arduinoblocks

Una vez creado sale en el cuadro de Active MQTT Credentials

Cluster

Billing

Help

Feedback

Credentials

OVERVIEW

ACCESS MANAGEMENT

INTEGRATIONS

WEB CLIENT

GETTING STARTED

Set up credentials for your IoT devices

Define the credentials that your MQTT clients can use to connect to your HiveMQ Cloud cluster.
Please visit the [HiveMQ documentation](#) for examples on how to use the credentials to connect an MQTT client to your cluster.
(All fields are mandatory)

Username

At least 5 characters. Username must be unique

Password

At least 8 characters, numbers, upper- and lowercase letters.

Confirm Password

Passwords must match.

ADD

Active MQTT Credentials

These credentials allow MQTT clients to publish and subscribe to your HiveMQ Cloud cluster.

Username	Password	Actions
	*****	DELETE

En Arduinoblocks

En Inicio

- Conexión wifi
- Iniciamos el broker con broker.hivemq.com
- Puerto 1883
- Usuario y contraseña las que has puesto en la ventana anterior
- De paso hemos inicializado el LCD para que nos muestre las variables que vamos a transmitir
- También de paso enviamos al TDR-STEAM reiniciar el topic **video** para que borre el contenido anterior

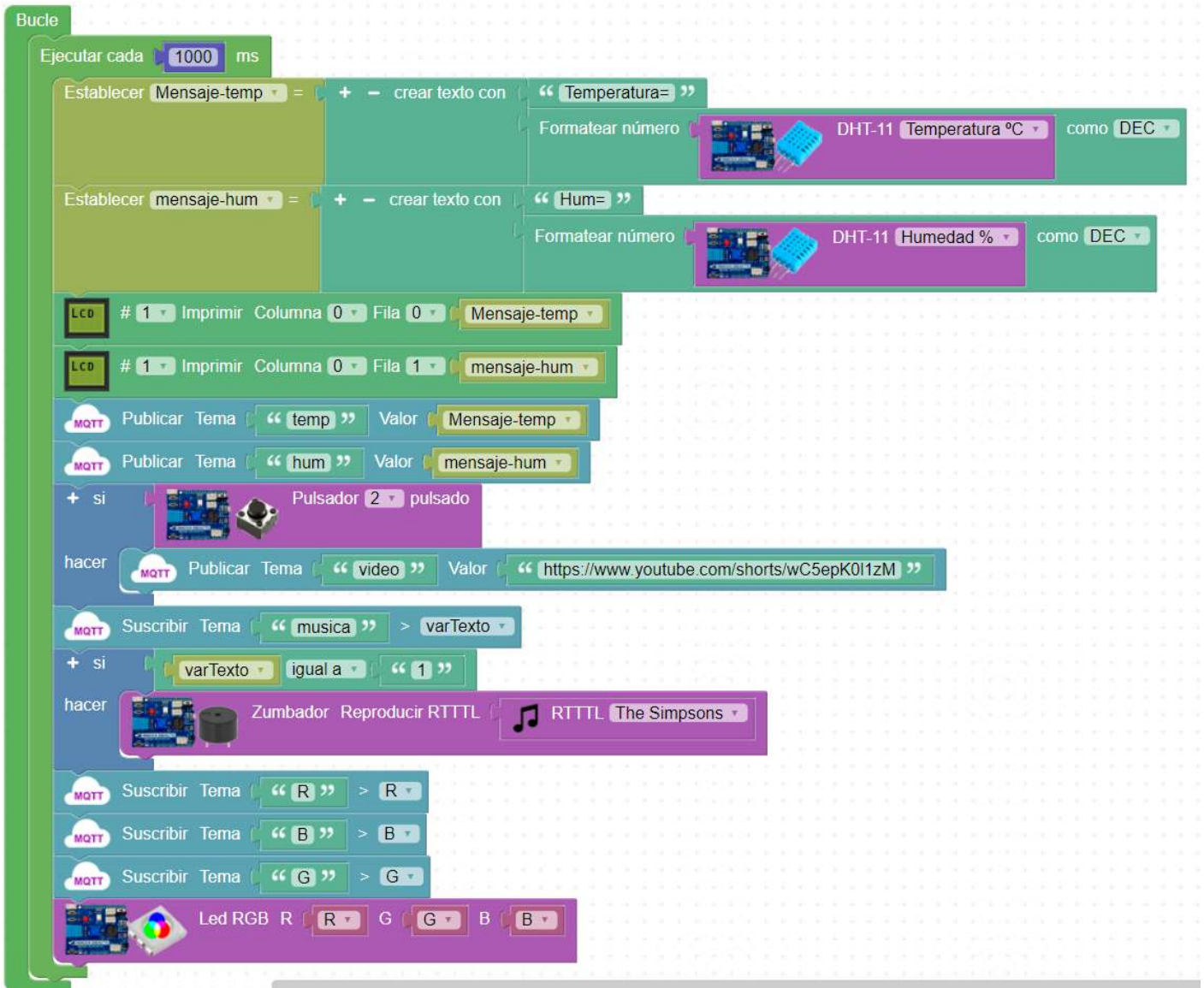


En bucle

- **DE TDR-STEAM AL CLIENTE**
 - Vamos a crear una variable texto **Mensaje-temp** que diga la temperatura
 - Idem con otra variable **Mensaje-hum**
 - Lo mostramos en el LCD para ver los mensajes
 - Lo publicamos en los topic de **temp** y **hum** respectivamente (nombres arbitrarios pero relacionados con las variables a mostrar)
 - Si se pulsa el pulsador 2, que en el topic **video** muestre la URL de un vídeo

• DE CLIENTE AL TDR-STEAM

- Suscribimos al topic **musica**
 - Si recibe un 1 pues que suene una música
- Suscripción a los tópicos **R, G y B** que se almacenarán en las variables correspondientes
 - Modificar el led RGB a los valores recibidos.



En cliente: Las APPs posibles

Vamos a usar una APP como cliente tanto para visualizar los valores que nos envía la TDR-STEAMaker como para enviarle desde la APP. Hay varias APP para trabajar los MQTT, vamos a ver varias

MyMQTT

Muy didáctica pero no muy visual

<https://play.google.com/store/apps/details?id=at.tripwire.mqtt.client>

Se trabaja de forma artesanal que viene bien para entender los mecanismos del protocolo MQTT:

- En la pestaña de **Subscribe** se van marcando los **topics** que se quieren visualizar en el **Dashboard**
 - El Dashboard simplemente visualiza los valores, no existen Gauges o displays que lo visualizan
- En la pestaña de **Publish** escribe el topic y el mensaje y se envía
 - No tiene pues la posibilidad de poner botones, sliders en el Dashboard para enviar, se envía como un texto

MyMQTT

instant:solutions OG

Contiene anuncios · Compras en aplicaciones

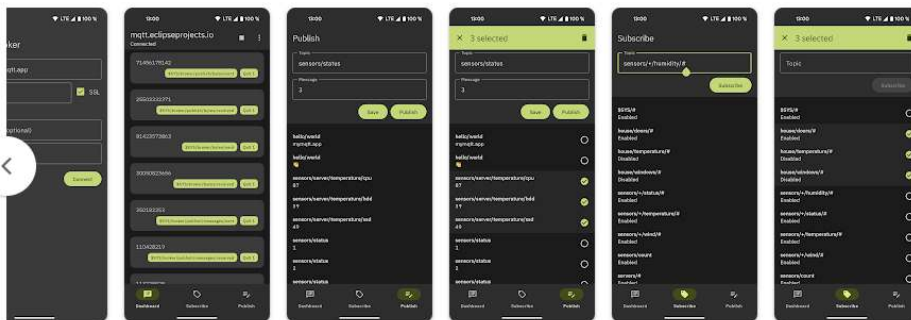
4,3★
811 reseñas

100 mil+
Descargas

PEGI 3

Instalar en más dispositivos

Esta aplicación está disponible para todos tus dispositivos



Información de contacto del desarrollador

Sitio web
<https://mymqtt.app>

Correo electrónico
feedback@instant-apps.at

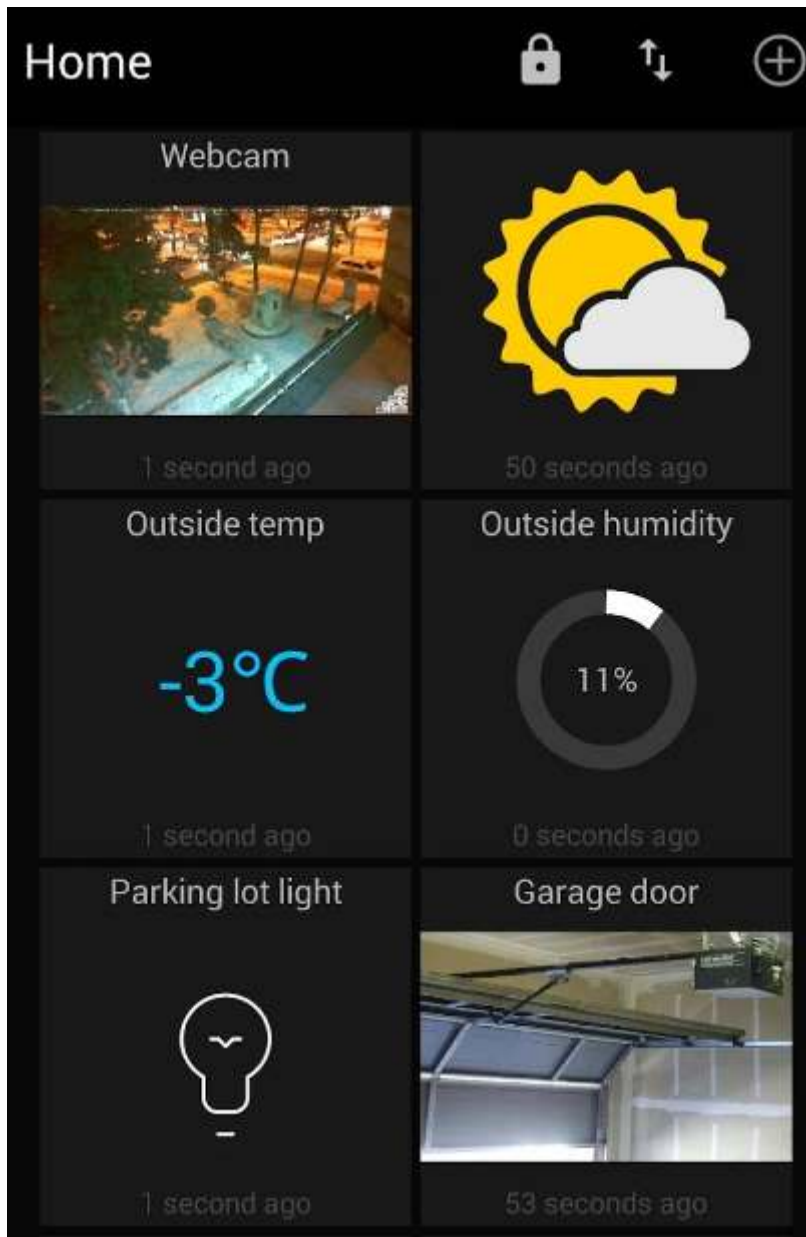
Dirección
Bauernstraße 1 4600 Wels Österreich

Política de privacidad

MQTT Dash

<https://play.google.com/store/apps/details?id=net.routix.mqttdash>

Está orientada a la domótica, tiene buena visualización pero no tiene todos los gadgets deseables para nuestra STEAM Maker



IoT MQTT Panel

<https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod>

Tiene buena visualización y buenos gadgets. Nos quedamos con esta para las siguientes páginas, pero recomendamos también practicar la MyMQTT

IoT MQTT Panel

Rahul Kundu

Contiene anuncios

4,6★
1,24 mil reseñas

100 mil+
Descargas

PEGI 3

Instalar en más dispositivos

Esta aplicación está disponible para todos tus dispositivos



Información de contacto del desarrollador

Sitio web
<http://www.snrlab.in/iot/iot-mqtt-panel-user-guide/>

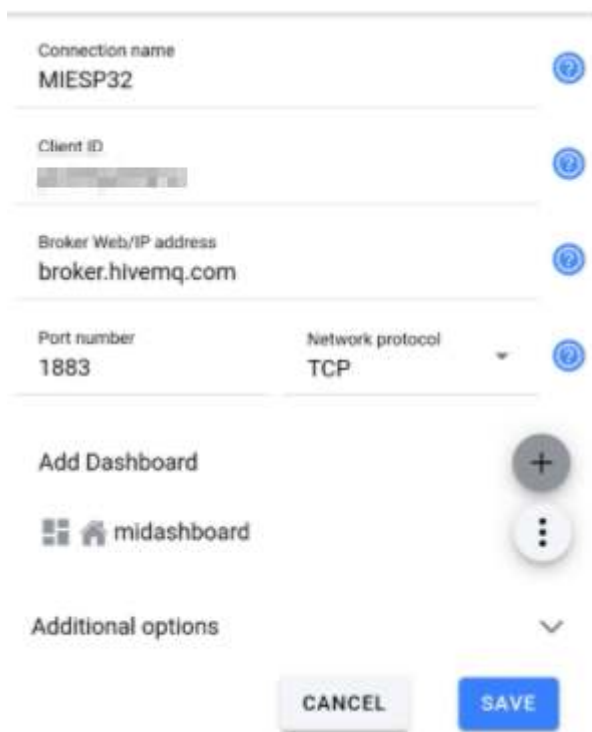
Correo electrónico
snrlab.in@gmail.com

Dirección
Rahul Kundu, Nikunjapur Bankura West Bengal
India

Actividad 17 APP IoT MQTT Panel

Conexión

Creamos una nueva conexión rellenando el nombre de nuestro **usuario** en hivemq creado en Active MQTT Credentials (ver <https://libros.catedu.es/books/esp32-en-el-aula/page/hivemq-crear-credenciales>) y de broker : **broker.hivemq.com**



The screenshot shows a configuration window for an MQTT connection. It includes the following fields and controls:

- Connection name:** MIESP32
- Client ID:** A text field with a generated ID (partially obscured).
- Broker Web/IP address:** broker.hivemq.com
- Port number:** 1883
- Network protocol:** TCP (selected from a dropdown menu)
- Add Dashboard:** A button with a plus icon.
- midashboard:** A button with a house icon.
- Additional options:** A dropdown menu with a downward arrow.
- CANCEL** and **SAVE** buttons at the bottom.

En Add Dashboard creamos nuestro panel de control. Puede haber varios paneles de control para la misma conexión-

Si te has fijado NO PIDE LA CONTRASEÑA cualquiera que tenga tu nombre de usuario puede interactuar con tu dispositivo. Por eso tal y como se ha dicho en <https://libros.catedu.es/books/esp32-en-el-aula/page/mqtt> **no son seguros, sólo utilizarlo para fines educativos y de pruebas**

Dashboard

Vamos añadiendo en el botón + los gadgets que queramos, son muy intuitivos, en la figura se ha puesto:

- Dos **Gauge** asociados a los topic **hum** y **temp** de valores mínimos 0 y 100
- Un **Switch** asociado al topic **musica**
 - En Payload on le hemos puesto el valor 1
 - En Payload off el valor 0
- Tres **sliders** asociados a los topics R, G y B con valores mínimos 0 y máximos 255
- Por último un gadget **URL Laucher** asociado al topic **video**



El resultado

https://www.youtube.com/embed/B_l-yk84ppU