

Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

Mosquitto

Eclipse Mosquitto™ es un servidor de mensajes de código abierto (con licencia EPL/EDL) que implementa las versiones 3.1 y 3.1.1 del protocolo MQTT. MQTT proporciona un método ligero para llevar a cabo la mensajería utilizando un modelo de publicación/suscripción. Esto lo hace adecuado para la “Internet de los objetos” de mensajería como con sensores de baja potencia o dispositivos móviles como teléfonos, ordenadores embebidos o microcontroladores como el Arduino.

Mosquitto es un broker MQTT OpenSource ampliamente utilizado debido a su ligereza lo que nos permite, fácilmente, emplearlo en gran número de ambientes, incluso si éstos son de pocos recursos.

Web: <https://mosquitto.org/>

Testear Mosquitto: <https://test.mosquitto.org/>

Repositorio: <https://github.com/eclipse/mosquitto>

A través de [esta página](#) puedes descargar y seguir los pasos de instalación según cual sea tu sistema operativo.

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies](#).

Cerrar y aceptar

<https://geekytheory.com/wp-content/uploads/2015/05/jerarquia.png>

Proyectos Eclipse Open Source for IoT

- <https://iot.eclipse.org/>
- <https://iot.eclipse.org/projects/>

Más información:

- Documentación: <https://mosquitto.org/documentation/>
- download: <https://mosquitto.org/download/>
- <https://www.eclipse.org/mosquitto/download/>
- MQTT comunity: <https://github.com/mqtt/mqtt.github.io/wiki>
- Mosquitto ayuda: <https://mosquitto.org/man/mqtt-7.html>
- Instalar Windows: <http://www.steves-internet-guide.com/install-mosquitto-broker/>

Otro MQTT broker: <http://www.hivemq.com/>

Para descargar e instalar mosquitto seguir las instrucciones de: <https://mosquitto.org/download/>

Buen tutorial de Luis del Valle: <https://programarfacil.com/esp8266/mqtt-esp8266-raspberry-pi/>

Instalar Mosquitto en Raspberry Pi

Mosquitto está disponible en el repositorio principal de Raspberry Pi. También hay repositorios Debian proporcionados por el proyecto mosquitto, tal y como se describe en <https://mosquitto.org/blog/2013/01/mosquitto-debian-repository/>

Lo primero es descargar la signing key o clave de firma utilizando el comando wget. Este comando descarga el fichero indicado como parámetro en el directorio en el que te encuentras.

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar

Añadimos la clave para a una lista para autenticar el paquete que vamos a descargar más tarde.

- `sudo apt-key add mosquitto-repo.gpg.key`

Después descargamos la lista de repositorios de Mosquitto con wget en la carpeta adecuada.

- `cd /etc/apt/sources.list.d/`
- `sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list` (depende la version de raspian que usemos)

<https://programarfacil.com/wp-content/uploads/2018/02/terminal-mosquitto-04.png>

Actualizamos la lista de paquetes disponibles y sus versiones

- `sudo apt-get update`

Y finalmente instalamos Mosquitto y los clientes

- `sudo apt-get install mosquitto`
- `sudo apt-get install mosquitto-clients`

<https://programarfacil.com/wp-content/uploads/2018/02/terminal-mosquitto-08.png>

Si va a utilizar MQTT en un proyecto de Python, tendrá que instalar paho-mqtt, que reemplaza al antiguo módulo de Mosquitto Python. Si python-pip no está instalado tendrá que instalarlo primero:

- `sudo apt-get install python-pip`
- `sudo pip install paho-mqtt`

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar

- <https://mosquitto.org/download/>
- <https://alexkychen.wordpress.com/2017/11/09/install-mosquitto-on-raspbian-stretch/>
- <https://theembeddedlab.com/tutorials/install-mosquitto-on-a-raspberry-pi/>
- <https://programarfacil.com/esp8266/mqtt-esp8266-raspberry-pi/>
- Getting started:
- <http://www.switchdoc.com/2016/02/tutorial-installing-and-testing-mosquitto-mqtt-on-raspberry-pi/>
- Tutorial: <https://www.instructables.com/id/Installing-MQTT-BrokerMosquitto-on-Raspberry-Pi/>
- Video: <https://www.youtube.com/watch?v=Y-H6grpWdec>

Configurar mosquito

La configuración de mosquitto está el fichero **/etc/mosquitto/mosquitto.conf**. Recordar hacer copia del fichero antes de hacer alguna modificación

Para añadir todos los mensajes de log en el fichero de log añadir las líneas:

```
# Save all log in file
```

```
log_dest file /var/log/mosquitto/mosquitto.log
```

```
log_type all
```

```
log_timestamp true
```

Para más información sobre las opciones del fichero mosquitto.conf ver </usr/share/doc/mosquitto/examples/mosquitto.conf>

Para reiniciar el servicio de mosquito usar:

- `sudo systemctl restart mosquitto`

En Linux puede recargar los archivos de configuración sin reiniciar el broker enviando la señal HUP de la siguiente manera:

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies](#).

Cerrar y aceptar

<http://www.steves-internet-guide.com/wp-content/uploads/send-hup-mosquitto.png>

Más información:

- <http://www.steves-internet-guide.com/mosquitto-conf-file/>
- <https://theembeddedlab.com/tutorials/install-mosquitto-on-a-raspberry-pi/>
- <http://www.switchdoc.com/2016/02/tutorial-installing-and-testing-mosquitto-mqtt-on-raspberry-pi/>

Comprobar Funcionamiento de Mosquitto

El último paso es probar nuestra instalación. Vamos a usar dos terminales. Uno se suscribirá al tema “test-mosquitto”, y el otro publicará un mensaje sobre este tema. La prueba tendrá éxito si el mensaje enviado por el editor se registra en el terminal de abonado.

Abrir un terminal en el ordenador con PuTTY y para suscribirse al topic “test-mosquitto” poner el comando:

- **mosquitto_sub -d -t 'test-mosquitto'** (d = debug mode, t = topic)

Las opciones del comando mosquitto_sub son: https://mosquitto.org/man/mosquitto_sub-1.html

Si quisiéramos conectarnos a otro mosquitto y no el de nuestra raspberry usar:

- **mosquitto_sub -d -h IP_BROKER -t 'test-mosquitto'** (d = debug mode, t = topic, h = host)

Abrir otro terminal y para publicar en el topic “test-mosquitto” poner el comando:

- **mosquitto_pub -d -t 'test-mosquitto' -m 'This is a test message'**

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar

https://i2.wp.com/theembeddedlab.com/wp-content/uploads/2018/01/test_message_received.png

Probar a suscribirse o publicar a otros mosquitto. También puedes hacerlo instalando el cliente MQTT.fx de: <https://mqttfx.jensd.de/>

Más información:

- <https://theembeddedlab.com/tutorials/install-mosquitto-on-a-raspberry-pi/>
- <https://programarfacil.com/esp8266/mqtt-esp8266-raspberry-pi/>
- <https://www.instructables.com/id/Installing-MQTT-BrokerMosquitto-on-Raspberry-Pi/>
- Testing un mosquitto server: <https://randomnerdtutorials.com/testing-mosquitto-broker-and-client-on-raspbberry-pi/>

Securizar Mosquitto – Autenticación y Autorización

Tenemos un servidor Mosquitto instalado y funcionando, pero cualquiera que pueda acceder al puerto 1883 de nuestra Raspberry Pi o servidor podrá publicar y suscribirse a topics y además los mensajes no están cifrados.

El objetivo es configurar un broker MQTT con autenticación para securizar un poco el acceso al mismo de forma que podamos exponerlo en un servidor público y aún así tengamos zonas privadas.

Vamos a configurar Mosquitto para que use contraseñas. Mosquitto incluye una utilidad para generar un archivo de contraseña especial llamado **mosquitto_passwd**. Este comando le pedirá que introduzca una contraseña para el nombre de usuario especificado y coloque los resultados en /etc/mosquitto/passwd. Ejecutar este comando y poner la contraseña.

- `sudo mosquitto_passwd -c /etc/mosquitto/passwd curso_iot`

Ahora abriremos un nuevo archivo de configuración para Mosquitto y le diremos que use este archivo de contraseñas para requerir inicios de sesión para todas las conexiones:

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar

Y escribir en el fichero:

```
password_file /etc/mosquitto/passwd
```

```
allow_anonymous false
```

allow_anonymous false deshabilitará todas las conexiones no autenticadas, y la línea del archivo password_file le indica a Mosquitto dónde buscar información de usuario y contraseña.

Una vez modificado el fichero reiniciar mosquitto:

- `sudo systemctl restart mosquitto`

En el directorio /etc/mosquitto/conf.d se guardan los ficheros de configuración adicionales.

Para publicar y suscribirse con usuario y contraseña usar:

- `mosquitto_pub -d -t "test" -m "hola_mundo" -u "curso_iot" -P "password"`
- `mosquitto_sub -d -t "test" -u "curso_iot" -P "password"`

Securizar Mosquitto – Cifrado SSL

Desafortunadamente, estamos enviando contraseñas sin encriptar a través de Internet. Lo arreglaremos añadiendo cifrado SSL a Mosquitto.

MQTT and TLS

MQTT se basa en el protocolo de transporte TCP. De forma predeterminada, las conexiones TCP no utilizan una comunicación cifrada. Para encriptar toda la comunicación MQTT, muchos Brokers MQTT (como Mosquitto) permiten el uso de TLS en lugar de TCP simple. Si utiliza los campos de nombre de usuario y contraseña del paquete MQTT CONNECT para los mecanismos de autenticación y autorización, debería considerar seriamente el uso de TLS.

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar

TLS Overhead

El uso de MQTT sobre TLS tiene sus inconvenientes: la seguridad tiene un costo en términos de uso de la CPU y gastos generales de comunicación. Mientras que el uso adicional de la CPU es típicamente insignificante en el broker, puede ser un problema para los dispositivos muy restringidos que no están diseñados para tareas de computación intensiva. Técnicas como la reanudación de sesión pueden mejorar drásticamente el rendimiento de TLS.

La sobrecarga de comunicaciones del Handshake de TLS puede ser significativa si se espera que las conexiones de los clientes MQTT sean de corta duración. Aunque la cantidad de ancho de banda que necesita el handshake TLS depende de muchos factores, algunas mediciones muestran que establecer una nueva conexión TLS puede requerir hasta unos pocos kilobytes de ancho de banda. Dado que cada paquete está encriptado en TLS, los paquetes en el cable tienen una sobrecarga adicional en comparación con los paquetes no encriptados.

Si utiliza conexiones TCP de larga duración con MQTT (¡lo que debería hacer!), la sobrecarga de TLS, especialmente la sobrecarga de handshake de TLS, puede ser insignificante. Si trata con muchas reconexiones y no puede utilizar la reanudación de sesión, entonces la sobrecarga puede ser significativa, especialmente si está publicando mensajes MQTT muy pequeños. Si cada byte del cable cuenta para su caso de uso, TLS puede no ser la mejor opción para usted debido a la sobrecarga. Asegúrese de medir cuánta sobrecarga de TLS crea para su caso de uso.

TLS en dispositivos limitados

A veces, TLS no es factible para dispositivos limitados debido a la insuficiencia de recursos. Dependiendo de las cifras utilizadas, TLS puede ser muy intensivo en el cálculo y puede requerir muchos kilobytes de memoria.

Para hacerlo seguir estos tutoriales:

- <https://mcuoneclipse.com/2017/04/14/enable-secure-communication-with-tls-and-the-mosquitto-broker/>
- <http://www.steves-internet-guide.com/mosquitto-tls/>
- <https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl>

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies](#).

Cerrar y aceptar

- Seguridad en mosquitto con usuario y password usando API HTTP: <https://ricveal.com/blog/mqtt-mosquitto-seguridad/>
- Montar un mosquitto seguro: <https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker-on-ubuntu-16-04>

Anuncios



Esta entrada se publicó en Curso IoT Open Source y está etiquetada con Curso IoT Open Source en 19 noviembre, 2018 [<https://aprendiendoarduino.wordpress.com/2018/11/19/mosquitto/>] .

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

☺

Privacidad & Cookies: este sitio usa cookies. Al continuar usando este sitio, estás de acuerdo con su uso. Para saber más, incluyendo como controlar las cookies, mira aquí: [Política de Cookies.](#)

Cerrar y aceptar