Instalación librería PAHO MQTT CPP en Windows

INDICE

[Pasos para la instalación 1](#_Toc58584672)

[Puntos importantes a recordar 6](#_Toc58584673)

[Integración de las librerías en la aplicación ODK 6](#_Toc58584674)

[Integracion de la aplicación ODK en TIA PORTAL 6](#_Toc58584675)

[Resultados 6](#_Toc58584676)

[Bibliografía 7](#_Toc58584677)

Para instalar la librería paho mqtt cpp necesitaremos varias herramientas:

* Visual Studio 🡪 Nosotros utilizaremos la versión 2019, pero es recomendable utilizar una posterior a 2015.
* Cmake 🡪 Para instalar paho.mqtt.cpp en Visual Studio.

Antes de empezar hay que tener en cuenta una cosa muy importante: la librería paho.mqtt.cpp **no funciona** sin paho.mqtt.c. Por lo tanto, primero hay que instalar la librería paho.mqtt.c.

# Pasos para la instalación

Como hemos comentado, lo primero es instalar la herramienta Visual Studio: <https://visualstudio.microsoft.com/es/downloads/> .

Lo primero que hay que hacer es instalar la herramienta **CMake**: <https://cmake.org/download/> (nosotros hemos escogido la opción *Windows win64-x64 Installer*). Descargamos el instalador e instalamos la herramienta en el ordenador.

A continuación, descargamos la librería paho.mqtt.c. Para ello, podremos descargar el ZIP desde **github** o si tenemos la herramienta GIT ejecutar git clone para descargar el repositorio: <https://github.com/eclipse/paho.mqtt.c.git>

Dentro de la carpeta que nos crea, la cual se llama *paho.mqtt.c*, creamos una carpeta que con el nombre ***build***. Aquí se crearán todos los archivos.

El siguiente paso es abrir **cmake**. En ella tendremos que poner la carpeta *paho.mqtt.c* en el apartado de **Browse Source…** , y la carpeta build dentro de ella en el apartado **Browse Build…**

Después haremos clic en el botón de **Configure** (los crearemos con la opción Win32, NO FUNCIONA PARA x64) y esperaremos a que se ejecute. Si ha ido todo bien nos aparecerá este mensaje en la consola:

*Selecting Windows SDK version 10.0.18362.0 to target Windows 10.0.19041.*

*The C compiler identification is MSVC 19.25.28614.0*

*Detecting C compiler ABI info*

*Detecting C compiler ABI info - done*

*Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.25.28610/bin/Hostx64/x64/cl.exe - skipped*

*Detecting C compile features*

*Detecting C compile features - done*

*CMake version: 3.18.5*

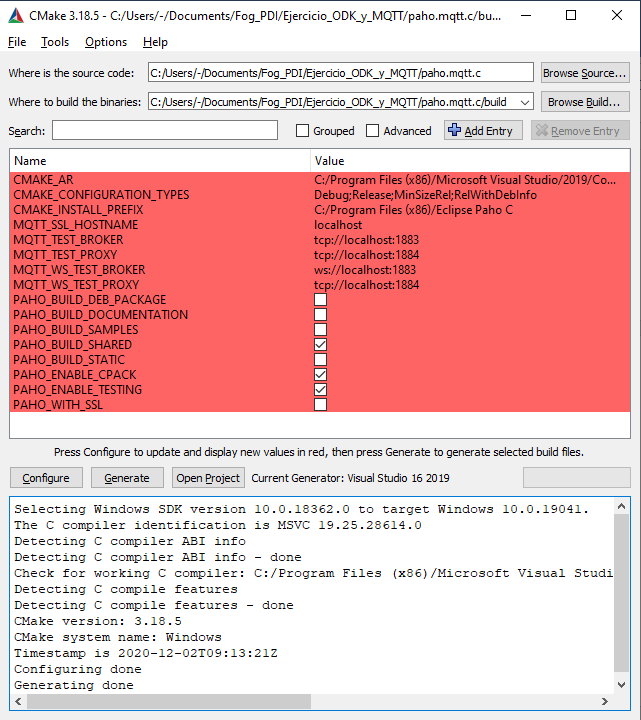
*CMake system name: Windows*

*Timestamp is 2020-12-02T09:13:21Z*

*Configuring done*

*Generating done*

Cuando acabe, haremos clic en el botón **Generate**.



A este momento de la instalación habremos acabado con cmake. Ahora, abriremos, en mi caso, el **Developer Command Prompt for VS 2019**, o la versión que tengan. Para ello podremos pulsar el botón Windows y buscar “developer”.

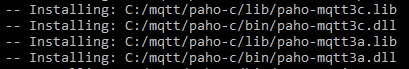
Dentro de ahí iremos hasta la carpeta *paho.mqtt.c*: 

y ejecutaremos los siguientes comandos:

> cmake -Bbuild -H. -DCMAKE\_INSTALL\_PREFIX=C:\mqtt\paho-c

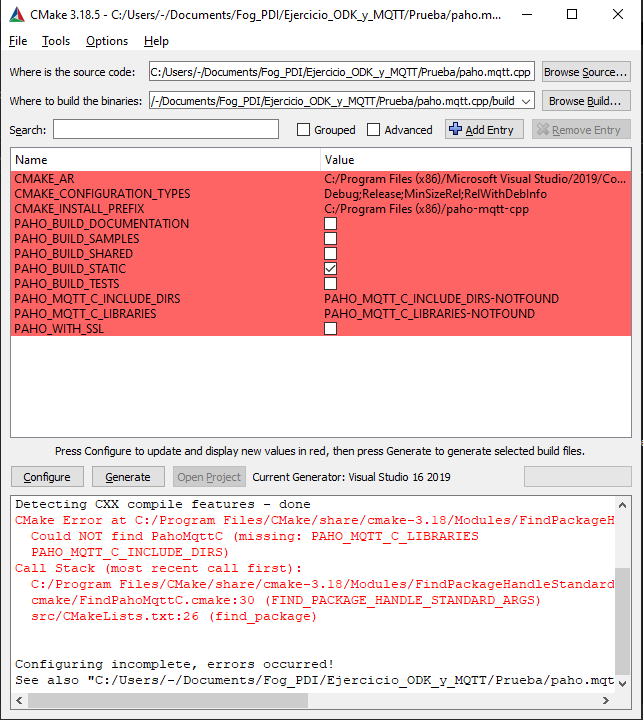
> cmake --build build/ --target install

Hay que fijarse en **donde** nos instala la librería:

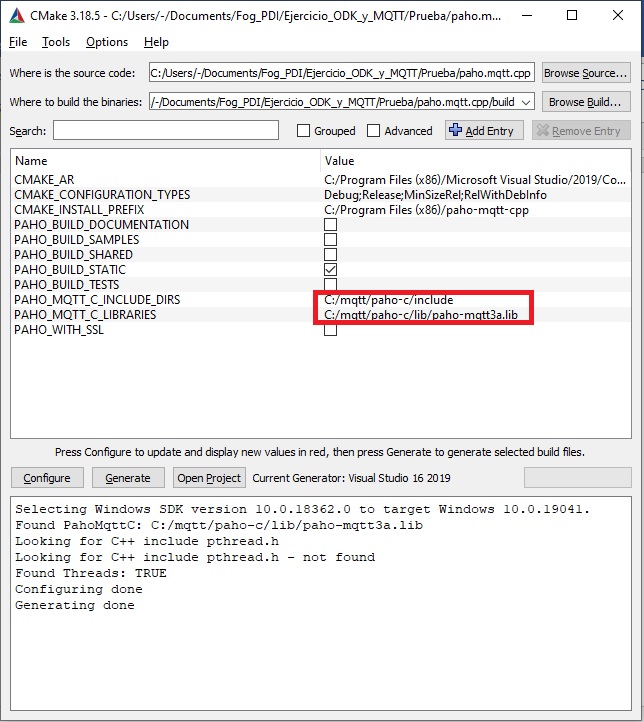


Teniendo la librería C construida, ahora lo haremos con la de C++. Para ello descargaremos también paho.mqtt.cpp como lo hemos hecho con C, desde <https://github.com/eclipse/paho.mqtt.cpp.git>. La carpeta que nos cree la dejaremos en la misma ruta que la de C.

Abriremos CMake, y repetiremos la acción como con la librería C, pondremos la ruta a *paho.mqtt.cpp* y a *paho.mqtt.cpp/build*. Nos dará este error:



Este error es porque no hemos añadido la librería c que se ha creado, para ello, tendremos que rellenar esta información.



Después, ejecutaremos estos comandos en el Developer Command Prompt for VS 2019 (suponiendo que seguimos en la carpeta de *paho.mqtt.c*):

> cd ..\paho.mqtt.cpp

> cmake -Bbuild -H. -DCMAKE\_INSTALL\_PREFIX=C:\mqtt\paho-cpp -DPAHO\_BUILD\_SAMPLES=ON -DPAHO\_WITH\_SSL=OFF -DCMAKE\_PREFIX\_PATH=C:\mqtt\paho-c

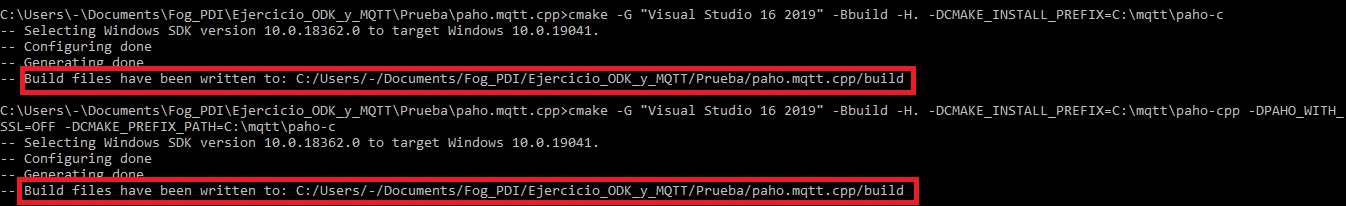
> cmake --build build/ --target install

Esto crea e instala las dos librerías a una ubicación dentro de *C:\mqtt*. (suele estar en el path de la variable CMAKE\_PREFIX\_PATH).

Por último, ejecutaremos estos dos comandos:

> cmake -G "Visual Studio 16 2019" -Bbuild -H. -DCMAKE\_INSTALL\_PREFIX=C:\mqtt\paho-c

> cmake -G "Visual Studio 16 2019" -Bbuild -H. -DCMAKE\_INSTALL\_PREFIX=C:\mqtt\paho-cpp -DPAHO\_WITH\_SSL=OFF -DCMAKE\_PREFIX\_PATH=C:\mqtt\paho-c



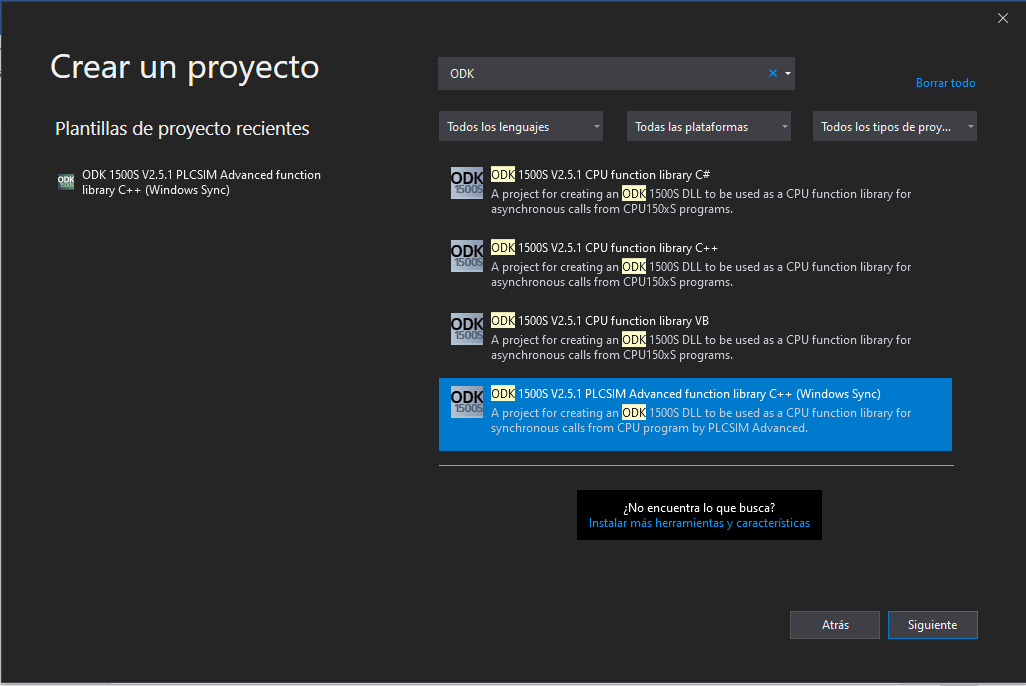
## Puntos importantes a recordar

Note that it is very important that you use the same generator (target) to build BOTH libraries, otherwise you will get lots of linker errors when you try to build the C++ library.

Now, you can reference both libraries in your c++ projects. Be aware that if you want to use the paho.mqtt.cpp library on your project you'll also have to include paho.mqtt.c.

# Integración de las librerías en la aplicación ODK

Una vez tengamos instaladas las dos librerías, podremos añadirlos a un proyecto ODK en Visual Studio. Para ello, crearemos el proyecto con una plantilla de Visual Studio:

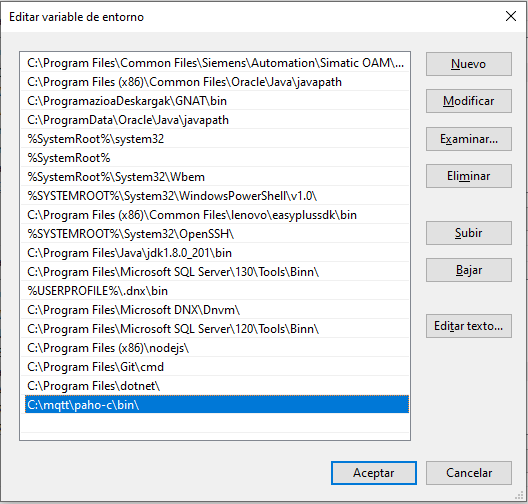


Después, tendremos que configurar el proyecto para integrar las librerías. Para ellos iremos a la pestaña Proyecto 🡪 Propiedades de <nombre del proyecto>. Ahí, tendremos que configurar dos apartados:

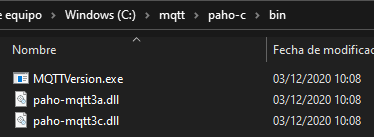
En C/C++ 🡪 General 🡪 Directorios de inclusión adicionales

# Integracion de la aplicación ODK en TIA PORTAL

Para que la aplicación ODK pueda encontrar la librería Paho MQTT tenemos que añadir la ruta a los archivos DLL en las variables del sistema del equipo:



Si la instalación se ha realizado correctamente, en esta ruta se deberán encontrar estos archivos:



# Resultados

# Bibliografía

* <https://github.com/eclipse/paho.mqtt.c>
* <https://github.com/eclipse/paho.mqtt.cpp/>
* <https://stackoverflow.com/questions/56455358/how-to-build-paho-mqtt-c-on-windows>
* <https://www.eclipse.org/paho/index.php?page=clients/cpp/index.php>
* <https://programmersought.com/article/79372757728/> (esta no he mirado pero la he encontrado luego)

Para el código dentro de la aplicación ODK:

* <https://cumulocity.com/guides/device-sdk/mqtt-examples/#hello-mqtt-c>