Example on compiling Azure-IoT-SDK-for-Embedded-C with the toolchain of WP76 module

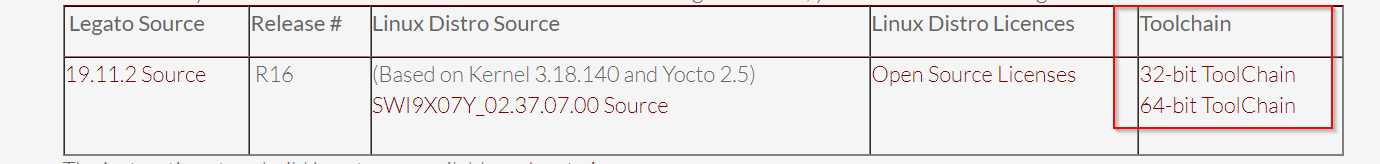
# Background

1. The Azure SDK for Embedded C is designed to allow small embedded (IoT) devices to communicate with Azure services.
2. The cross-compilation example below is done with Ubuntu 16 with CMAKE version 3.5.1
3. The version of Azure-IoT-SDK-for-Embedded-C in this document is 1.11.0.

# Procedure

1. Download and install the toolchain in /opt/swi according to your module's FW. For example, below is the toolchain of firmware R16:

<https://source.sierrawireless.com/resources/airprime/software/wp76xx/wp76xx-firmware-latest-release-components/>



1. Download the main branch of Azure-IoT-SDK-for-Embedded-C source code to “Source” folder.

mkdir Source

cd Source

git clone --recursive https://github.com/Azure/azure-iot-sdk-c.git

1. Put the following toolchain-rpi.cmake to ./azure-iot-sdk-c/build\_all/linux



1. Starting from Azure-IoT-SDK-for-Embedded-C tag version 1.9.0, in ./azure-iot-sdk-c/CMakeLists.txt, around line 237, we need to add the following before "# Turn off tests for dependencies #":

# if any compiler has a command line switch called "OFF" then it will need special care

if (NOT "${compileOption\_C}" STREQUAL "")

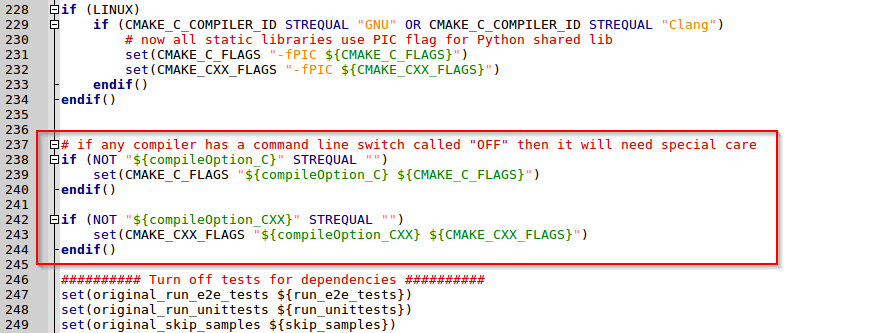
set(CMAKE\_C\_FLAGS "${compileOption\_C} ${CMAKE\_C\_FLAGS}")

endif()

if (NOT "${compileOption\_CXX}" STREQUAL "")

set(CMAKE\_CXX\_FLAGS "${compileOption\_CXX} ${CMAKE\_CXX\_FLAGS}")

endif()

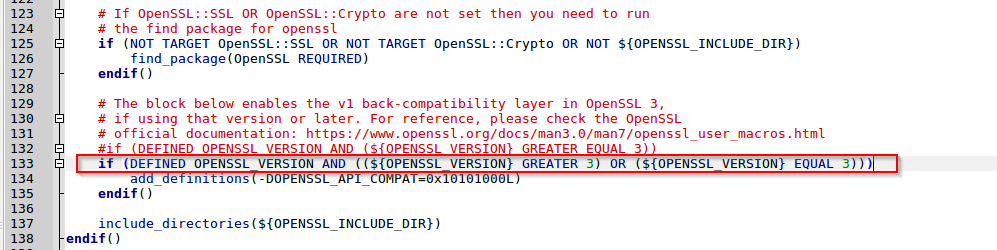


1. If your CMAKE version is smaller than 3.7, there is no "GREATER\_EQUAL" arguments implementation in CMakeLists.txt, we need to manually change "GREATER\_EQUAL" to "GREATER OR EQUAL" argument in "c-utility/CMakeLists.txt", around line 132:

if (DEFINED OPENSSL\_VERSION AND ((${OPENSSL\_VERSION} GREATER 3) OR (${OPENSSL\_VERSION} EQUAL 3)))

add\_definitions(-DOPENSSL\_API\_COMPAT=0x10101000L)

endif()



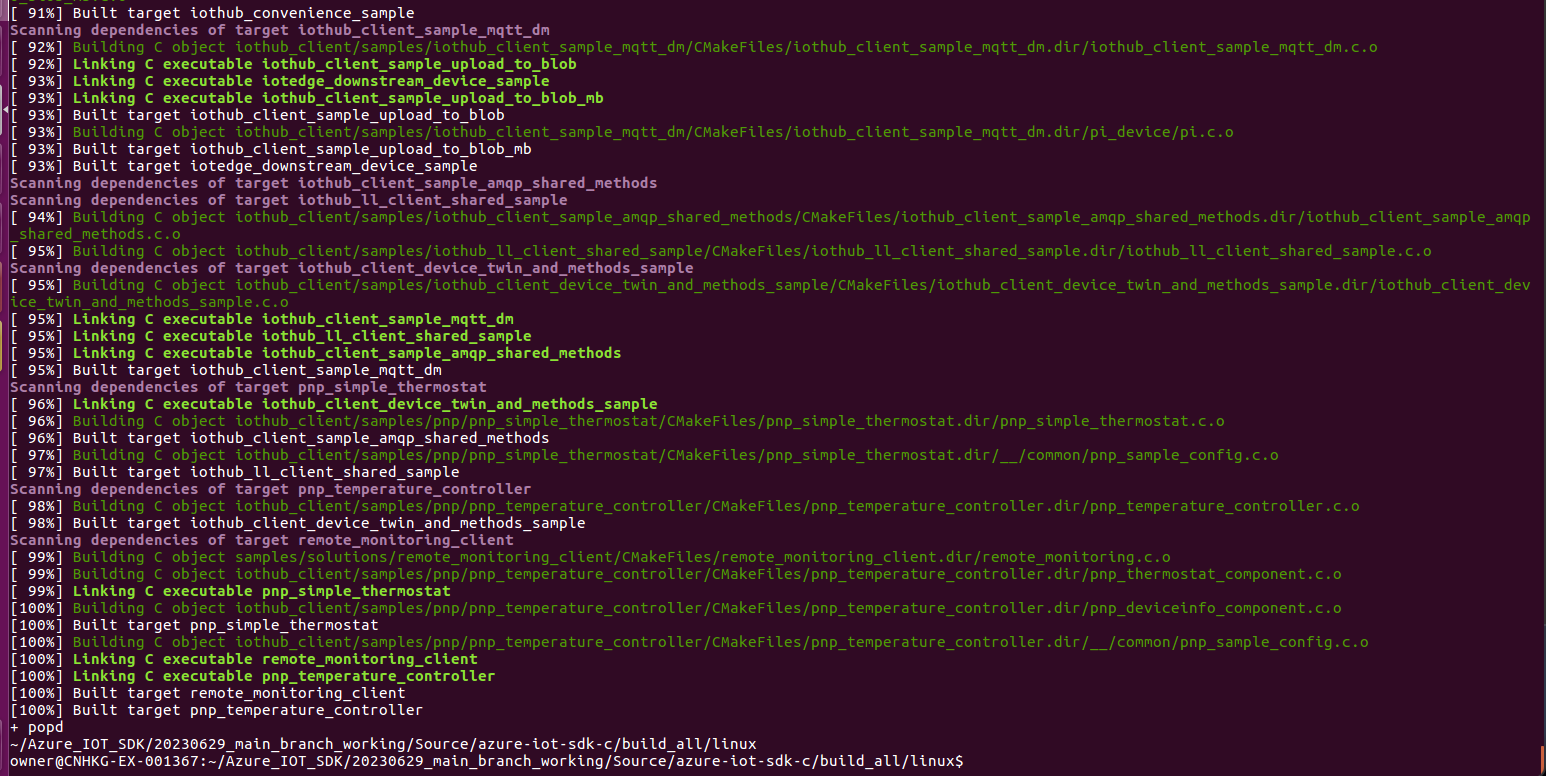
1. Type the following to cross compile the SDK with toolchain of WP76 module according to your build environment:

export RPI\_ROOT=/opt/swi/SWI9X07Y\_02.37.07.00/sysroots/armv7a-neon-poky-linux-gnueabi

cd ./azure-iot-sdk-c/build\_all/linux

./build.sh --toolchain-file toolchain-rpi.cmake -cl --sysroot=$RPI\_ROOT

1. If compile successfully, you should see the terminal like this:



1. Transfer the sample binary (e.g. ./azure-iot-sdk-c/cmake/ samples/solutions/remote\_monitoring\_client/remote\_monitoring\_client) to /tmp folder of module by SCP tool
2. Type the following in WP76 module to run the sample binary:

root@swi-mdm9x28-wp:/tmp# chmod 777 remote\_monitoring\_client

root@swi-mdm9x28-wp:/tmp# ./remote\_monitoring\_client

This sample simulates a Chiller device connected to the Remote Monitoring solution accelerator

Creating IoTHub handle

Error: Time:Sun Jan 6 08:23:17 1980 File:/home/owner/Azure\_IOT\_SDK/20230629\_main\_branch\_working/Source/azure-iot-sdk-c/iothub\_client/src/iothub\_client\_core\_ll.c Func:IoTHubClientCore\_LL\_CreateFromConnectionString Line:1417 Tokenizer error

Error: Time:Sun Jan 6 08:23:17 1980 File:/home/owner/Azure\_IOT\_SDK/20230629\_main\_branch\_working/Source/azure-iot-sdk-c/iothub\_client/src/iothub\_client\_core\_ll.c Func:IoTHubClientCore\_LL\_CreateFromConnectionString Line:1555 iotHubName is not found

Error: Time:Sun Jan 6 08:23:17 1980 File:/home/owner/Azure\_IOT\_SDK/20230629\_main\_branch\_working/Source/azure-iot-sdk-c/iothub\_client/src/iothub\_client\_core.c Func:create\_iothub\_instance Line:1115 Failure creating iothub handle

Failure creating IotHub device. Hint: Check your connection string.