# cyclonedds官方示例编译为arm可执行文件

arm框架开发板不具备自主编译源代码的能力，所以需要在x86 linux 系统下进行预编译后移植至arm框架后才能运行。以下所有操作是在PC的ubuntu 虚拟机中进行的（编译方法待完善）。

## 安装gcc和g++交叉编译工具链

1. 执行安装命令：

sudo apt-get update

sudo apt-get install gcc-arm-linux-gnueabihf

sudo apt-get install g++-arm-linux-gnueabihf

sudo apt-get update

sudo apt-get install -y gcc-aarch64-linux-gnu g++-aarch64-linux-gnu cmake

安装后工具位于：/usr/bin目录下。

su@su-virtual-machine:~$ which aarch64-linux-gnu-gcc

/usr/bin/aarch64-linux-gnu-gcc

su@su-virtual-machine:~$ which aarch64-linux-gnu-g++

/usr/bin/aarch64-linux-gnu-g++

1. 检查glibc版本：

在开发板中查看glibc版本：：

root@topeet:/$ strings /lib/aarch64-linux-gnu/ libc.so.6 |grep GLIBC\_

GLIBC\_2.17

GLIBC\_2.18

GLIBC\_2.22

GLIBC\_2.23

GLIBC\_2.24

GLIBC\_2.25

GLIBC\_2.26

GLIBC\_2.27

GLIBC\_2.28

GLIBC\_2.29

GLIBC\_2.30

GLIBC\_PRIVATE

在虚拟机中查看glibc版本：

su@su-virtual-machine:/$ strings /usr/aarch64-linux-gnu/lib/libc.so.6 |grep GLIBC\_

GLIBC\_2.17

GLIBC\_2.18

GLIBC\_2.22

GLIBC\_2.23

GLIBC\_2.24

GLIBC\_2.25

GLIBC\_2.26

GLIBC\_2.27

GLIBC\_2.28

GLIBC\_2.29

GLIBC\_2.30

GLIBC\_PRIVATE

如果ubuntu中交叉编译链的glibc中有比目标机更高的版本，则可能出现生成可执行文件运行时报错缺失库。（交叉编译链中位于/usr/aarch64-linux-gnu）  
 解决：因为glibc与内核相关，并不能通过简单的一条指令就可以升级，建议降低ubuntu交叉编译链glibc版本。

## 生成/install\_DDS\_ARM（arm版本DDS库）

1. 拷贝和解压工程文件：

拷贝 到ubuntu虚拟机下新建的~/CYCLONEDDS/ 目录下并解压:

sudo apt-get install unrar

sudo apt-get install unzip

unzip cyclonedds-master.zip -d ./cyclonedds-master-x86

unzip cyclonedds-master.zip -d ./cyclonedds-master-arm

1. 新建arm.cmake文件：

在cyclonedds-master目录下新建arm.cmake文件

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /**cyclonedds-master**$ ls

**arm.cmake** cmake colcon.pkg CYCLONEDDS\_QUALITY\_DECLARATION.md examples LICENSE package.xml README.md SECURITY.md

arm.cmake 内容如下：

# the name of the target operating system

set(CMAKE\_SYSTEM\_NAME Linux)

set(CMAKE\_SYSTEM\_PROCESSOR arm)

# which C and C++ compiler to use

set(CMAKE\_C\_COMPILER aarch64-linux-gnu-gcc)

set(CMAKE\_CXX\_COMPILER aarch64-linux-gnu-g++)

# where is the target environment

set(CMAKE\_FIND\_ROOT\_PATH /usr/aarch64-linux-gnu)

# adjust the default behavior of the FIND\_XXX() commands:

# search headers and libraries in the target environment, search

# programs in the host environment

set(CMAKE\_FIND\_ROOT\_PATH\_MODE\_PROGRAM NEVER)

set(CMAKE\_FIND\_ROOT\_PATH\_MODE\_LIBRARY ONLY)

set(CMAKE\_FIND\_ROOT\_PATH\_MODE\_INCLUDE ONLY)

1. 新建build.sh 文件：

在cyclonedds-master目录下新建build.sh文件

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /**cyclonedds-master**$ ls

arm.cmake **build.sh** CMakeCPack.cmake compat docs fuzz NOTICE.md PkgConfig.pc.in ROADMAP.md src

azure-pipelines.yml CHANGELOG.rst CMakeLists.txt CONTRIBUTING.md etc hooks PackageConfig.cmake.in ports scripts WiX

build-arm cmake colcon.pkg CYCLONEDDS\_QUALITY\_DECLARATION.md examples LICENSE package.xml README.md SECURITY.md

build.sh内容如下：

sudo rm -rf build-arm

sudo mkdir -p build-arm

cd build-arm

sudo cmake -DCMAKE\_TOOLCHAIN\_FILE=../arm.cmake -DCMAKE\_INSTALL\_PREFIX=/install\_DDS\_ARM -DENABLE\_SSL=NO -DCMAKE\_PREFIX\_PATH=/install\_DDS\_ARM ..

sudo cmake --build .

sudo cmake --build . --target install

1. 生成arm版本DDS库**install\_DDS\_ARM**：

编译过程编通过执行[build.sh](https://build.sh) 完成

build.sh添加执行权限（chmod +x build.sh）

执行 sudo ./build.sh进行编译arm版本DDS库

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm$ sudo ./build.sh

-- The C compiler identification is GNU 9.4.0

System is unknown to cmake, create:

-- Installing: /install\_DDS\_ARM/include/dds/ddsc/dds\_internal\_api.h

-- Installing: /install\_DDS\_ARM/include/dds/ddsc/dds\_basic\_types.h

-- Installing: /install\_DDS\_ARM/include/dds/dds.h

-- Up-to-date: /install\_DDS\_ARM/include

-- Up-to-date: /install\_DDS\_ARM/include/dds

-- Installing: /install\_DDS\_ARM/include/dds/export.h

-- Installing: /install\_DDS\_ARM/lib/libddsc.so

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm$ ls

arm.cmake build.sh CMakeCPack.cmake compat docs fuzz NOTICE.md PkgConfig.pc.in ROADMAP.md src

azure-pipelines.yml CHANGELOG.rst CMakeLists.txt CONTRIBUTING.md etc hooks PackageConfig.cmake.in ports scripts WiX

build-arm cmake colcon.pkg CYCLONEDDS\_QUALITY\_DECLARATION.md examples LICENSE package.xml README.md SECURITY.md

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm$ cd /

su@su-virtual-machine:/$ ls

bin boot cdrom dev etc home **install\_DDS\_ARM** lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv swapfile sys tmp usr var

su@su-virtual-machine:/$

ARM版的DDS库会自动生成到根目录下。

## 生成/install\_DDS\_x86（x86版本DDS库）

编译x86版本的DDS库和应用程序不需要arm.cmake

1. 新建build.sh：

cyclonedds-master-x86目录下新建build.sh，内容如下：

mkdir -p build

cd build

sudo cmake -DCMAKE\_INSTALL\_PREFIX=/install\_DDS\_x86 -DENABLE\_SSL=NO -DCMAKE\_PREFIX\_PATH=/install\_DDS\_x86 ..

sudo cmake --build .

sudo cmake --build . --target install

1. 生成下x86版本DDS库**install\_DDS\_x86**：

添加build.sh执行权限，执行build.sh

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-x86$ sudo ./build.sh

如果/install\_DDS\_x86目录存在，则表示./build.sh执行成功。

## 生成下x86版本应用程序

1. 新建build.sh：

在cyclonedds-master-x86/cyclonedds-master/examples/helloworld/目录下，新建build.sh：

build.sh内容如下：

mkdir -p build

cd build

sudo cmake -DENABLE\_SSL=NO -DCMAKE\_PREFIX\_PATH=/install\_DDS\_x86 ..

sudo make

1. 生成下x86 版本可执行程序：

在cyclonedds-master/examples/helloworld应用程序目录下，添加执行权限，执行sudo ./build.sh 编译x86版本可执行程序：

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-x86/cyclonedds-master/examples/helloworld$ sudo ./build.sh

-- Building internal IDLC backend

-- Configuring done

-- Generating done

-- Build files have been written to: /home/su/CYCLONEDDS/cyclonedds-master-x86/cyclonedds-master/examples/helloworld/build

Scanning dependencies of target HelloWorldData\_lib\_generate

[ 11%] Generating HelloWorldData.c, HelloWorldData.h

[ 11%] Built target HelloWorldData\_lib\_generate

Scanning dependencies of target HelloworldSubscriber

[ 22%] Building C object CMakeFiles/HelloworldSubscriber.dir/subscriber.c.o

[ 33%] Building C object CMakeFiles/HelloworldSubscriber.dir/HelloWorldData.c.o

[ 44%] Linking C executable HelloworldSubscriber

[ 55%] Built target HelloworldSubscriber

Scanning dependencies of target HelloworldPublisher

[ 66%] Building C object CMakeFiles/HelloworldPublisher.dir/publisher.c.o

[ 77%] Building C object CMakeFiles/HelloworldPublisher.dir/HelloWorldData.c.o

[ 88%] Linking C executable HelloworldPublisher

[100%] Built target HelloworldPublisher

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-x86/cyclonedds-master/examples/helloworld$ ls

build build.sh CMakeLists.txt HelloWorldData.idl publisher.c readme.rst subscriber.c

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master/examples/helloworld$ cd ~/CYCLONEDDS/cyclonedds-master-x86/cyclonedds-master/examples/helloworld/build/

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-x86/cyclonedds-master/examples/helloworld/build$ ls

CMakeCache.txt CMakeFiles cmake\_install.cmake **HelloWorldData.c HelloWorldData.h** HelloworldPublisher HelloworldSubscriber Makefile

## 生成下ARM版本应用程序

生成arm工程应用程序编译需要来自x86工程应用目录下的.c 和.h文件：HelloWorldData.c HelloWorldData.h ，用来生成最终arm版本可执行文件**HelloworldPublisher HelloworldSubscriber**

1. 拷贝HelloWorldData.c 、 HelloWorldData.h ：

将x86工程里的HelloWorldData.c 、 HelloWorldData.h拷贝到arm工程文件目录cyclonedds-master-arm/examples/helloworld下：

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm/examples/helloworld$ ls

build build.sh CMakeLists.txt **HelloWorldData.c HelloWorldData.h** HelloWorldData.idl publisher.c readme.rst subscriber.c

1. 修改CMakeLists.txt：：

修改CMakeLists.txt内容如下：

#

# Copyright(c) 2019 to 2022 ZettaScale Technology and others

#

# This program and the accompanying materials are made available under the

# terms of the Eclipse Public License v. 2.0 which is available at

# http://www.eclipse.org/legal/epl-2.0, or the Eclipse Distribution License

# v. 1.0 which is available at

# http://www.eclipse.org/org/documents/edl-v10.php.

#

# SPDX-License-Identifier: EPL-2.0 OR BSD-3-Clause

#

cmake\_minimum\_required(VERSION 3.16)

project(helloword LANGUAGES C)

if(NOT TARGET CycloneDDS::ddsc)

# Find the CycloneDDS package.

find\_package(CycloneDDS REQUIRED)

endif()

# This is a convenience function, provided by the CycloneDDS package,

# that will supply a library target related the the given idl file.

# In short, it takes the idl file, generates the source files with

# the proper data types and compiles them into a library.

#idlc\_generate(TARGET HelloWorldData\_lib FILES "HelloWorldData.idl" WARNINGS no-implicit-extensibility)

# Both executables have only one related source file.

add\_executable(HelloworldPublisher publisher.c **HelloWorldData.c HelloWorldData.h**)

add\_executable(HelloworldSubscriber subscriber.c **HelloWorldData.c HelloWorldData.h**)

# Both executables need to be linked to the idl data type library and

# the ddsc API library.

target\_link\_libraries(HelloworldPublisher CycloneDDS::ddsc)

target\_link\_libraries(HelloworldSubscriber CycloneDDS::ddsc)

1. 新建build.sh：

在cyclonedds-master-arm/cyclonedds-master/examples/helloworld/目录新建build.sh

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /cyclonedds-master/examples/helloworld$ ls

build **build.sh** CMakeLists.txt HelloWorldData.c HelloWorldData.h HelloWorldData.idl publisher.c readme.rst subscriber.c

build.sh的内容如下：

sudo rm -rf build/

sudo mkdir build

cd build/

sudo cmake -DCMAKE\_PREFIX\_PATH=/install\_DDS\_ARM -DCMAKE\_TOOLCHAIN\_FILE=../../../arm.cmake ..

sudo make

1. 生成可执行文件

执行sudo ./build.sh，生成可执行文件HelloworldPublisher HelloworldSubscriber

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /cyclonedds-master/examples/helloworld$ sudo ./build.sh

-- The C compiler identification is GNU 9.4.0

-- Check for working C compiler: /usr/bin/aarch64-linux-gnu-gcc

-- Check for working C compiler: /usr/bin/aarch64-linux-gnu-gcc -- works

-- Detecting C compiler ABI info

-- Detecting C compiler ABI info - done

-- Detecting C compile features

-- Detecting C compile features - done

-- Configuring done

-- Generating done

-- Build files have been written to: /home/su/CYCLONEDDS/cyclonedds-master-arm /cyclonedds-master/examples/helloworld/build

Scanning dependencies of target HelloworldSubscriber

[ 16%] Building C object CMakeFiles/HelloworldSubscriber.dir/subscriber.c.o

[ 33%] Building C object CMakeFiles/HelloworldSubscriber.dir/HelloWorldData.c.o

[ 50%] Linking C executable HelloworldSubscriber

[ 50%] Built target HelloworldSubscriber

Scanning dependencies of target HelloworldPublisher

[ 66%] Building C object CMakeFiles/HelloworldPublisher.dir/publisher.c.o

[ 83%] Building C object CMakeFiles/HelloworldPublisher.dir/HelloWorldData.c.o

[100%] Linking C executable HelloworldPublisher

[100%] Built target HelloworldPublisher

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /cyclonedds-master/examples/helloworld$ cd build

su@su-virtual-machine:~/CYCLONEDDS/cyclonedds-master-arm /cyclonedds-master/examples/helloworld/build$ file Helloworld\*

**HelloworldPublisher**: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=deebcb3541284f6212c77106049bf3b5832d2641, for GNU/Linux 3.7.0, not stripped

**HelloworldSubscriber**: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=470464fd95f3db685df9ef724290a8a19efdcdca, for GNU/Linux 3.7.0, not stripped

## 编译步骤总结

大致步骤为：

1、安装gcc和g++工具链

2、生成/install\_DDS\_ARM

3、生成/install\_DDS\_x86

4、生成x86下的HelloWorldData.c HelloWorldData.h

5、生成HelloworldPublisher HelloworldSubscriber

6、完成

## 编译方法参考链接

参考链接：

<https://blog.csdn.net/qq_42595610/article/details/131694658?ops_request_misc=&request_id=&biz_id=102&utm_term=cyclonedds%20arm&utm_medium=distribute.pc_search_result.none-task-blog-2~blog~sobaiduweb~default-0-131694658.142>

<https://blog.csdn.net/qq_42595610/article/details/131635561>

https://blog.csdn.net/pony12/article/details/132940617?spm=1001.2101.3001.6650.6&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-6-132940617-blog-131694658.235%5Ev43%5Econtrol&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-6-132940617-blog-131694658.235%5Ev43%5Econtrol&utm\_relevant\_index=13

<https://blog.csdn.net/wu10188/article/details/86542418>