



OpenCV 介绍、国内团队介绍、编译及运行、CMake系统介绍

软件所智能软件中心PLCT实验室郑志文实习生

目 录

01 OpenCV 介绍

02 国内团队介绍

03 编译及运行

04 CMake系统介绍

1 什么是OpenCV ?

OpenCV是一个基于BSD许可（开源）发行的**跨平台计算机视觉库**，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 **C 函数和少量 C++ 类**构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

OpenCV的用途：**提高计算机视觉项目开发的效率**

2 如何理解OpenCV ?

OpenCV:开放源代码计算机视觉库



2 如何理解OpenCV ?

- ◎ 是Intel公司贡献出来的，俄罗斯工程师团队贡献大部分C/C++带代码。
- ◎ 在多数图像处理相关的应用程序中被采用，BSD许可，可以免费应用在商业和研究领域
- ◎ 最新版本是OpenCV4.2.0 ,当前SDK支持语言包括了Java、Python、IoS和Android版本。



Vadim Pisarevsky

01 OpenCV 介绍

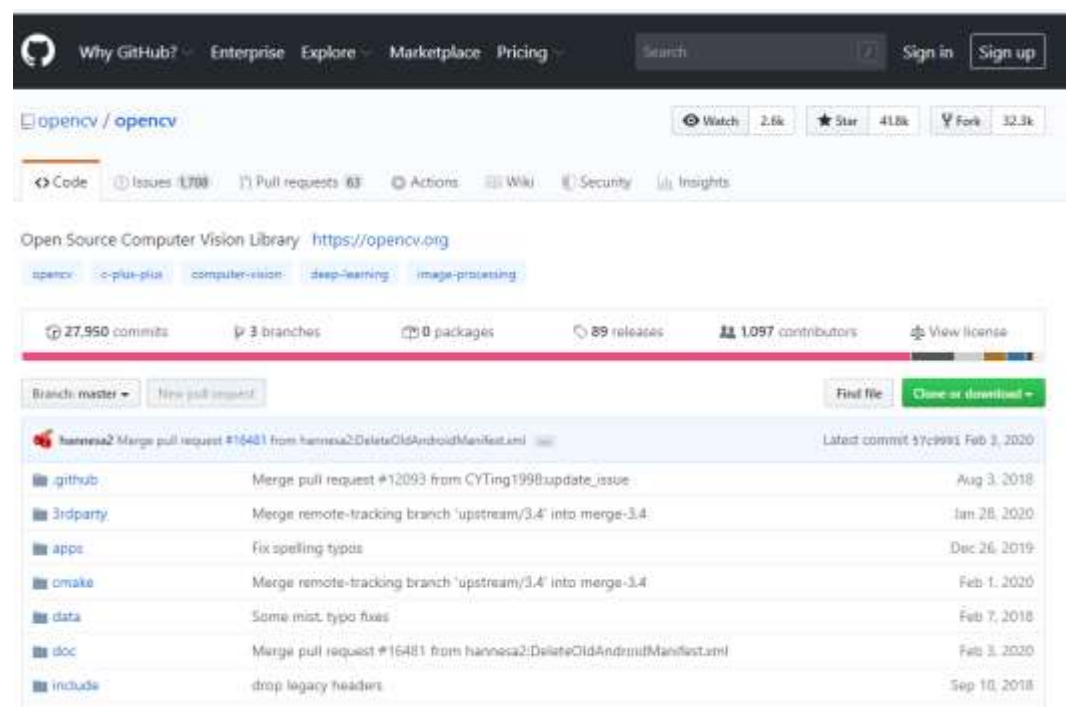
2 如何理解OpenCV ?

OpenCV官网 : <https://opencv.org/>

OpenCV的项目地址 : <https://github.com/opencv/opencv>



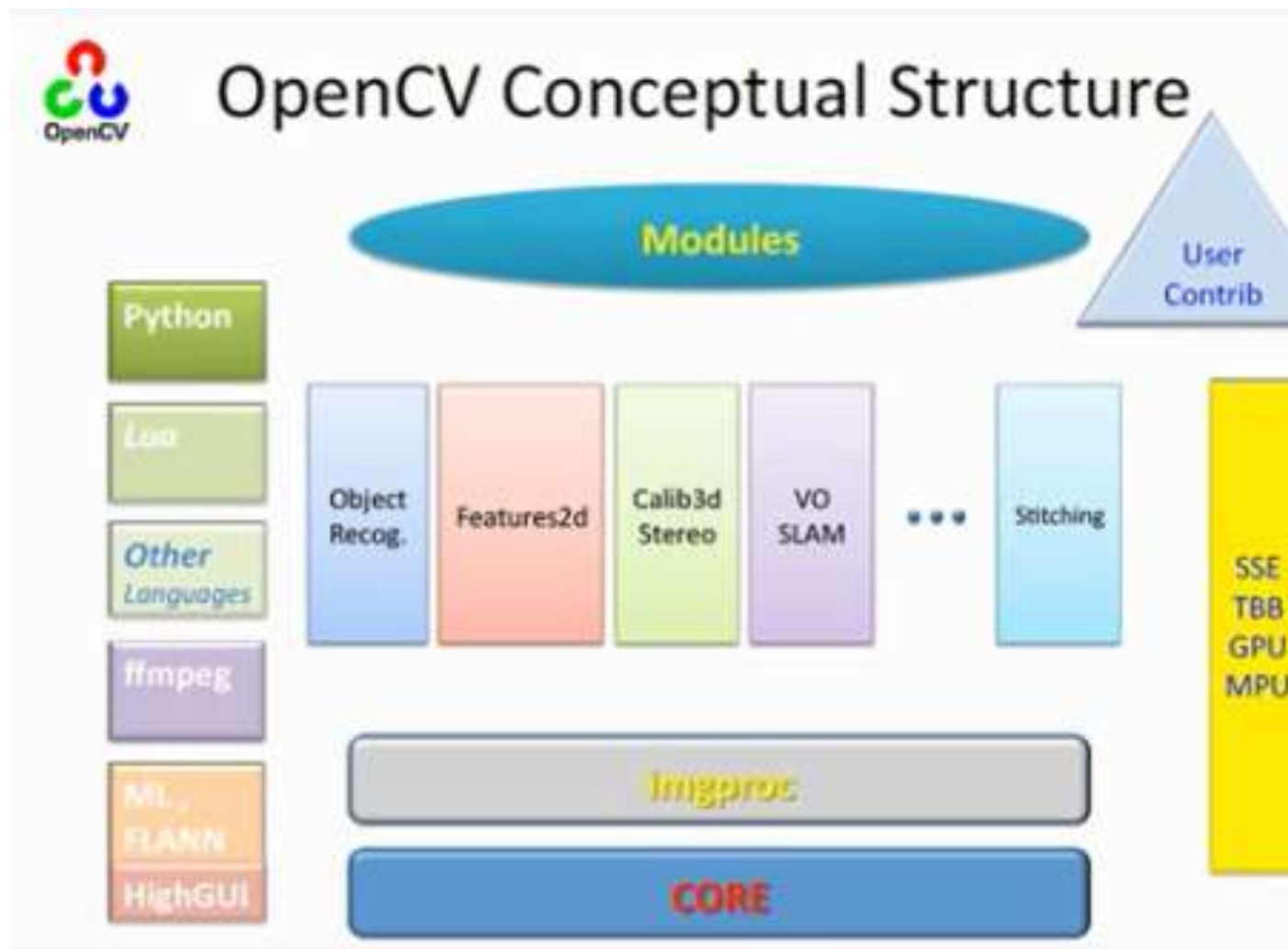
OpenCV官网



OpenCV项目地址

01 OpenCV 介绍

3 OpenCV的软件的架构及源代码介绍













OpenCV框架图

3 OpenCV的软件的架构及源代码介绍

核心模块






















- HighGU部分
- Image Process
- 2D Feature
- Camera Calibration and 3D reconstruction
- Video Analysis
- Object Detection
- Machine Learning
- GPU加速

3 OpenCV的软件的架构及源代码介绍

名称	修改日期	类型	大小
 3rdparty	2020/2/4 17:12	文件夹	
 apps	2020/2/4 17:12	文件夹	
 cmake	2020/2/4 17:12	文件夹	
 data	2020/2/4 17:12	文件夹	
 doc	2020/2/4 17:13	文件夹	
 include	2020/2/4 17:14	文件夹	
 modules	2020/2/4 17:15	文件夹	
 platforms	2020/2/4 17:15	文件夹	
 samples	2020/2/4 17:15	文件夹	
 .editorconfig	2020/2/3 21:34	EDITORCONFIG ...	1 KB
 CMakeLists.txt	2020/2/3 21:34	Text Document	64 KB
 CONTRIBUTING.md	2020/2/3 21:34	MD 文件	1 KB
 LICENSE	2020/2/3 21:34	文件	3 KB
 README.md	2020/2/3 21:34	MD 文件	1 KB
 SECURITY.md	2020/2/3 21:34	MD 文件	4 KB

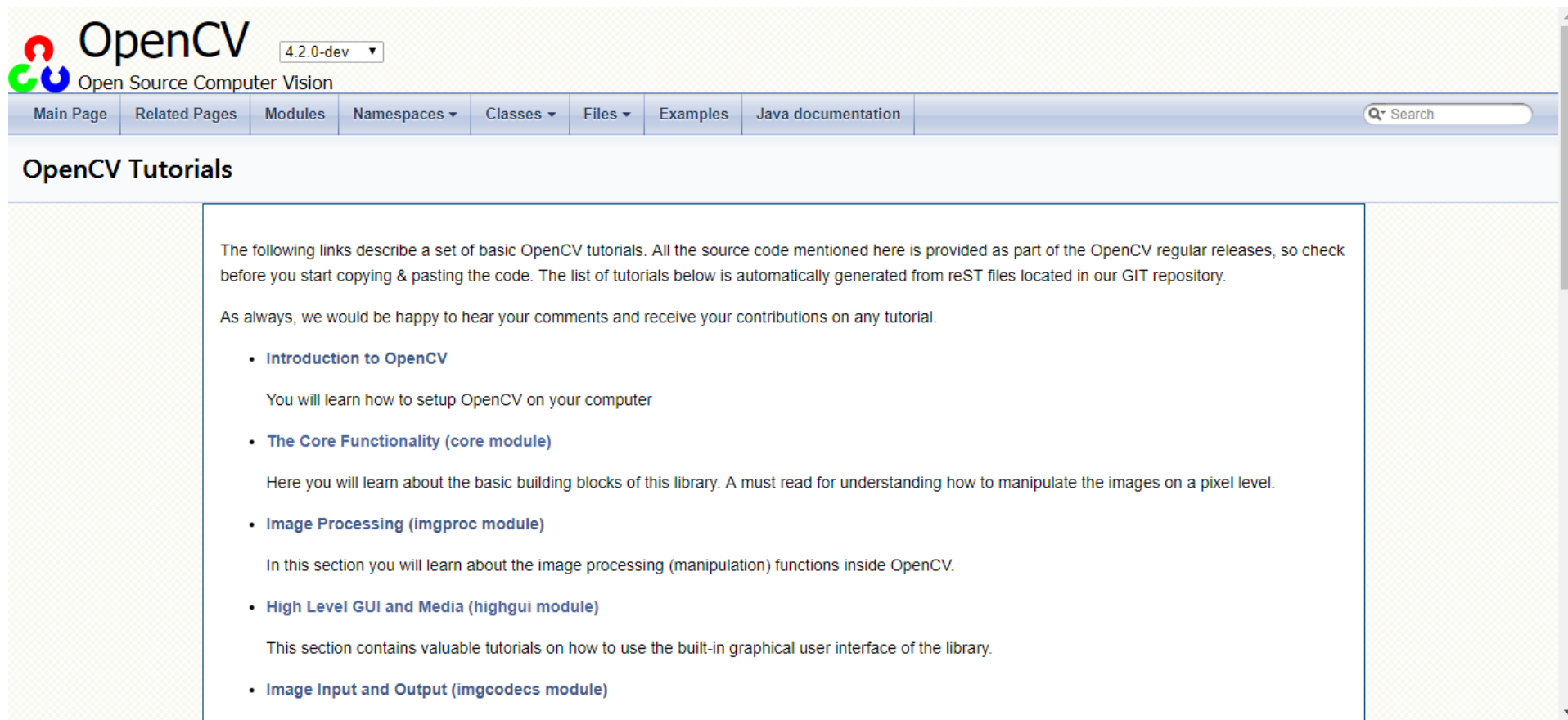
3 OpenCV的软件的架构及源代码介绍



此电脑 > 文件 (E:) > opencv-master > modules >			
名称	修改日期	类型	大小
 <u>calib3d</u>	2019/12/30 4:53	文件夹	
 <u>core</u>	2019/12/30 4:53	文件夹	
 <u>dnn</u>	2019/12/30 4:53	文件夹	
 <u>features2d</u>	2019/12/30 4:53	文件夹	
 <u>flann</u>	2019/12/30 4:53	文件夹	
 <u>gapi</u>	2019/12/30 4:53	文件夹	
 <u>highgui</u>	2019/12/30 4:53	文件夹	
 <u>imgcodecs</u>	2019/12/30 4:53	文件夹	
 <u>imgproc</u>	2019/12/30 4:53	文件夹	
 <u>java</u>	2019/12/30 4:53	文件夹	
 <u>js</u>	2019/12/30 4:53	文件夹	
 <u>ml</u>	2019/12/30 4:53	文件夹	
 <u>objdetect</u>	2019/12/30 4:53	文件夹	
 <u>photo</u>	2019/12/30 4:53	文件夹	
 <u>python</u>	2019/12/30 4:53	文件夹	
 <u>stitching</u>	2019/12/30 4:53	文件夹	
 <u>ts</u>	2019/12/30 4:53	文件夹	
 <u>video</u>	2019/12/30 4:53	文件夹	
 <u>videoio</u>	2019/12/30 4:53	文件夹	
 <u>world</u>	2019/12/30 4:53	文件夹	
 CMakeLists.txt	2019/12/30 4:53	文本文档	2 KB

3 OpenCV的软件的架构及源代码介绍

tutorials https://docs.opencv.org/master/d9/df8/tutorial_root.html

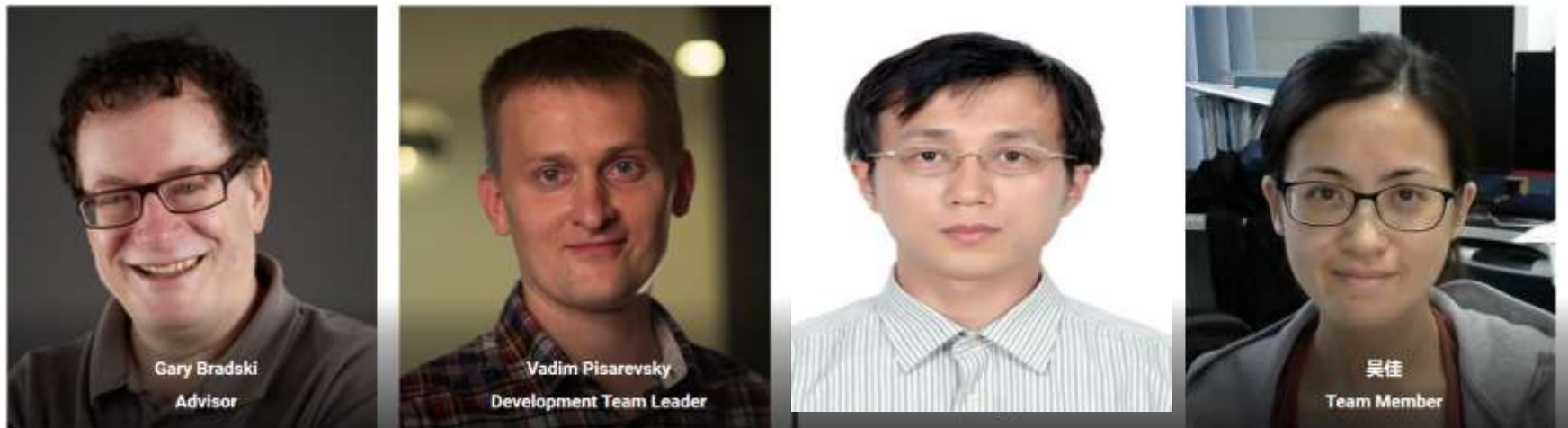


The screenshot displays the OpenCV documentation website. At the top, the OpenCV logo is visible next to the text 'OpenCV' and 'Open Source Computer Vision'. A dropdown menu shows '4.2.0-dev'. Below this is a navigation bar with links: 'Main Page', 'Related Pages', 'Modules', 'Namespaces', 'Classes', 'Files', 'Examples', and 'Java documentation'. A search bar is located on the right side of the navigation bar. The main content area is titled 'OpenCV Tutorials'. It contains a paragraph explaining that the following links describe a set of basic OpenCV tutorials, all of which are provided as part of the OpenCV regular releases. It also mentions that the list of tutorials is automatically generated from reST files located in the GIT repository. Below this paragraph, there is a list of tutorial topics, each with a brief description:

- **Introduction to OpenCV**
You will learn how to setup OpenCV on your computer
- **The Core Functionality (core module)**
Here you will learn about the basic building blocks of this library. A must read for understanding how to manipulate the images on a pixel level.
- **Image Processing (imgproc module)**
In this section you will learn about the image processing (manipulation) functions inside OpenCV.
- **High Level GUI and Media (highgui module)**
This section contains valuable tutorials on how to use the built-in graphical user interface of the library.
- **Image Input and Output (imgcodecs module)**

02 国内团队介绍

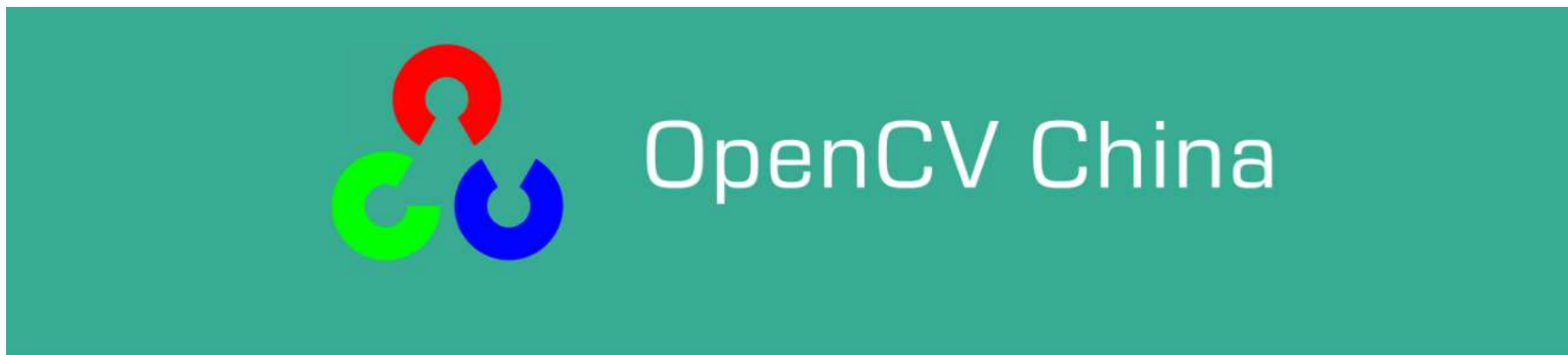
OpenCV中国团队于2019年9月成立，非营利目的，致力于OpenCV的开发、维护和推广工作。OpenCV中国团队由[深圳市人工智能与机器人研究院](#)支持，OpenCV项目发起人Gary Bradski担任团队顾问，OpenCV开发团队负责人**Vadim Pisarevsky**已全职加入，OpenCV中文社区创始人**于仕琪**博士担任团队负责人，团队成员吴佳、贾志刚等人。



团队主要成员

02 国内团队介绍

opencv中文网站 <http://www.opencv.org.cn/>



论坛

Forum



课程

Courses



合作

Collaboration

中文论坛

因为spam攻击及日益严格的网络监管，本论坛关闭发帖功能，仅供浏览。带来不便非常抱歉。



OpenCV
中文网站

 用QQ帐号登录
只需一步，快速开始

用户名

密码

☐ 自动登录

找回密码

登录

立即注册

首页 论坛 在线课程(英文) 新人入门文档 中文教程 中文文档(翻译中) 下载 Wiki

帖子

热搜: 安装 配置

论坛 OpenCV 技术 OpenCV中国团队

版块导航

OpenCV 技术

OpenCV中国团队

OpenCV新兵

视觉算法与理论

嵌入式平台开发

中文文档翻译

C#-OpenCV

OpenCV 其他

论坛管理



关注本站公众号

OpenCV中国团队 今日: 0 | 主题: 2 | 排名: 9

收藏本版 | 订阅

发帖

返回

全部主题	最新	热门	热帖	精华	更多	新窗	作者	回复/查看	最后发表
	注册规则改变：仅能通过QQ帐号注册本论坛 ... 2 3						Shiqi Yu 2015-10-26	26 14392	Thunder_wind 2019-11-12 09:06
	免费、高性能的人脸检测库 @ ... 2 3 4 5 6 .. 33						Shiqi Yu 2015-2-5	321 177192	dongdongdongdon 2019-3-27 14:30
版块主题									
	OpenCV.org在线课程众筹						Shiqi Yu 2019-5-26	1 2353	jsxyheu2014 2019-10-31 20:41

OpenCV中国团队技术负责人Vadim Pisarevsky受邀于2020年1月13日在中国开放指令生态（RISC-V）联盟2019年会上作报告，首次公开介绍OpenCV中国团队计划。



报告现场

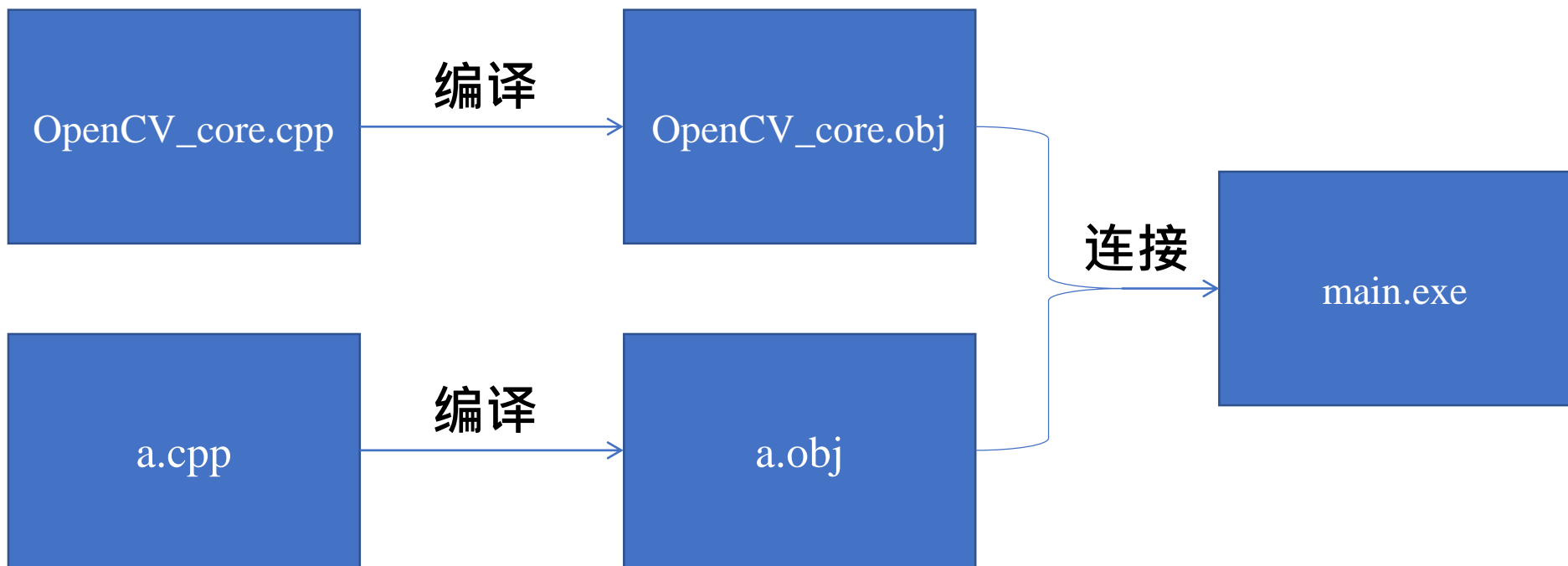
2020年OpenCV的计划与展望 OpenCV in 2020: Plans and Desires

- 2020年6月：OpenCV20周年及OpenCV 5发布
20-year anniversary and OpenCV 5 release – 2020 June
- OpenCV与边缘计算：针对嵌入式平台的优化与测试
OpenCV "at the edge" – optimization & testing on embedded platforms
 - 已与Open AI Lab合作，提升DNN在ARM上的运行速度
Joint project with OpenAI lab to accelerate DNN on ARM
- 增加对RISC-V的全面支持
RISC-V – first-class support in OpenCV
 - 诚邀各界合作，制定RISC-V的向量指令集标准，并将之应用到OpenCV中
Interested to work together with compiler teams, China RISC-V Alliance and companies to finalize RISC-V vector extension & intrinsics; integrate it into OpenCV
 - 将基于RISC-V的硬件纳入OpenCV持续集成系统（OpenCV CI System）
Put RISC-V hardware into OpenCV CI
- 加速和“标准化”深度学习：
Accelerate and "Standardize" Deep Learning:
 - 使OpenCV DNN支持更多类型NPU
Support various NPUs in OpenCV DNN
 - 与ONNX团队合作，增加对新网络的支持
Work with ONNX team to extend the specs
- 增加对3D传感器（光学雷达、ToF相机等）的支持
Add support for 3D sensors (lidars, ToF cameras ...)



报告PPT

OpenCV由一系列 C 函数和少量 C++ 类构成，当我们在使用时需要先对其进行编译。



我们以RISC-V交叉编译为例，讲解一下OpenCV的编译过程。

RISC-V，是一个基于精简指令集（RISC）原则的开源指令集架构（ISA）。

RISC-V交叉编译工具具有两种:针对裸机的和针对具有操作系统的开发环境。

针对裸机 `riscv64-unknown-elf-gcc`

针对具有操作系统的 `riscv64-unknown-linux-gnu-gcc`

我们以RISC-V交叉编译为例，讲解一下OpenCV的编译过程。

我们使用针对linux的RISC-V交叉编译工具

安装toolchain

1. 下载交叉编译工具：

```
$ git clone --recursive https://github.com/riscv/riscv-gnu-toolchain
```

2. 安装运行库

```
$ sudo apt-get install autoconf automake autotools-dev curl libmpc-dev libmpfr-  
dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc  
zlib1g-dev
```

3.配置安装路径

```
$ ./configure --prefix=/opt/riscv
```

4.编译(linux版本)

```
$ make linux
```

03 编译及运行

```
zww@zww-ThinkCentre-M828z-D101: ~/riscv-tools/riscv-gnu-toolchain
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
-I.././riscv-gcc/gcc/./libcpp/include -I.././riscv-gcc/gcc/./libdecnum
ber -I.././riscv-gcc/gcc/./libdecnumber/dpd -I../libdecnumber -I.././risc
v-gcc/gcc/./libbacktrace -o tree-ssa-propagate.o -MT tree-ssa-propagate.o -MM
D -MP -MF ../deps/tree-ssa-propagate.TPo .././riscv-gcc/gcc/tree-ssa-propagat
e.c
g++ -fno-PIE -c -g -O2 -DIN_GCC -DCROSS_DIRECTORY_STRUCTURE -fno-exceptions
-fno-rtti -fasynchronous-unwind-tables -W -Wall -Wno-narrowing -Wwrite-strings
-Wcast-qual -Wmissing-format-attribute -Woverloaded-virtual -pedantic -Wno-long-
long -Wno-variadic-macros -Wno-overlength-strings -DHAVE_CONFIG_H -I. -I. -I..
/./riscv-gcc/gcc -I.././riscv-gcc/gcc/. -I.././riscv-gcc/gcc/./include
-I.././riscv-gcc/gcc/./libcpp/include -I.././riscv-gcc/gcc/./libdecnum
ber -I.././riscv-gcc/gcc/./libdecnumber/dpd -I../libdecnumber -I.././risc
v-gcc/gcc/./libbacktrace -o tree-ssa-reassoc.o -MT tree-ssa-reassoc.o -MMD -M
P -MF ../deps/tree-ssa-reassoc.TPo .././riscv-gcc/gcc/tree-ssa-reassoc.c
g++ -fno-PIE -c -g -O2 -DIN_GCC -DCROSS_DIRECTORY_STRUCTURE -fno-exceptions
-fno-rtti -fasynchronous-unwind-tables -W -Wall -Wno-narrowing -Wwrite-strings
-Wcast-qual -Wmissing-format-attribute -Woverloaded-virtual -pedantic -Wno-long-
long -Wno-variadic-macros -Wno-overlength-strings -DHAVE_CONFIG_H -I. -I. -I..
/./riscv-gcc/gcc -I.././riscv-gcc/gcc/. -I.././riscv-gcc/gcc/./include
-I.././riscv-gcc/gcc/./libcpp/include -I.././riscv-gcc/gcc/./libdecnum
ber -I.././riscv-gcc/gcc/./libdecnumber/dpd -I../libdecnumber -I.././risc
v-gcc/gcc/./libbacktrace -o tree-ssa-sccvn.o -MT tree-ssa-sccvn.o -MMD -MP -M
F ../deps/tree-ssa-sccvn.TPo .././riscv-gcc/gcc/tree-ssa-sccvn.c
```

```
zww@zww-ThinkCentre-M828z-D101: ~/riscv-tools/riscv-gnu-toolchain
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
CXX dwarf2-frame-tailcall.o
CXX dwarf2-frame.o
CXX dwarf2expr.o
CXX dwarf2loc.o
CXX dwarf2read.o
CXX elfread.o
CXX eval.o
CXX event-loop.o
CXX event-top.o
CXX exceptions.o
CXX exec.o
CXX expprint.o
CXX extension.o
YACC f-exp.c
CXX f-exp.o
CXX f-lang.o
CXX f-typeprint.o
CXX f-valprint.o
CXX filename-seen-cache.o
CXX filesystem.o
CXX findcmd.o
CXX findvar.o
CXX frame-base.o
```

交叉编译工具链编译过程

5. 验证交叉编译工具：

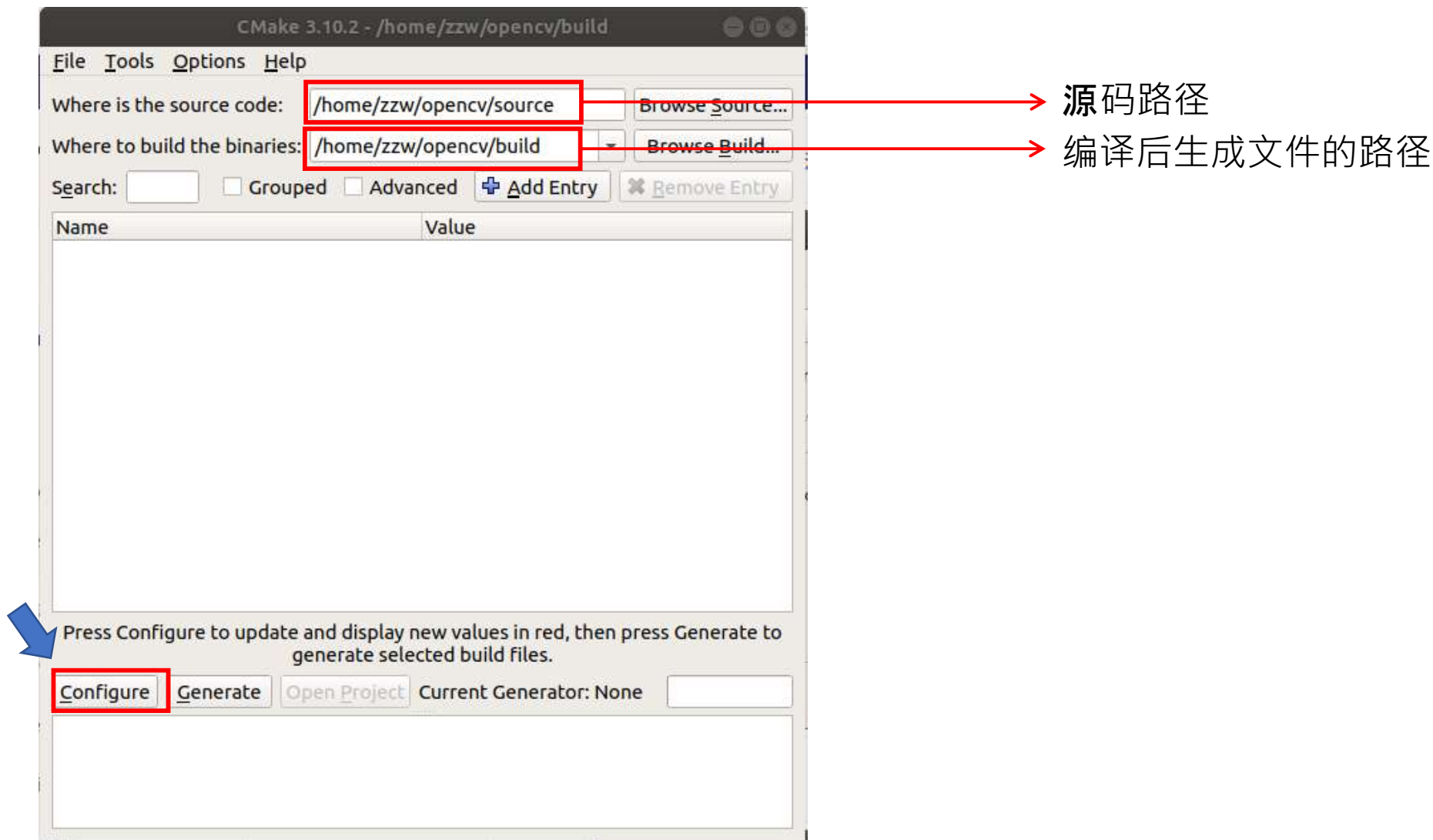
```
$ riscv64-unknown-linux-gnu-gcc -static hello.c -o hello
```

```
zzw@zzw-ThinkCentre-M828z-D101: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
zzw@zzw-ThinkCentre-M828z-D101:~$ riscv64-unknown-linux-gnu-gcc -static hello.c  
-o hello  
zzw@zzw-ThinkCentre-M828z-D101:~$ ls  
adobe-release-x86_64-1.0-1.noarch.rpm  hello.c      公共的  图片  音乐  
examples.desktop                     Python-2.7.15 模板    文档  桌面  
hello                                riscv-tools   视频    下载  
zzw@zzw-ThinkCentre-M828z-D101:~$
```

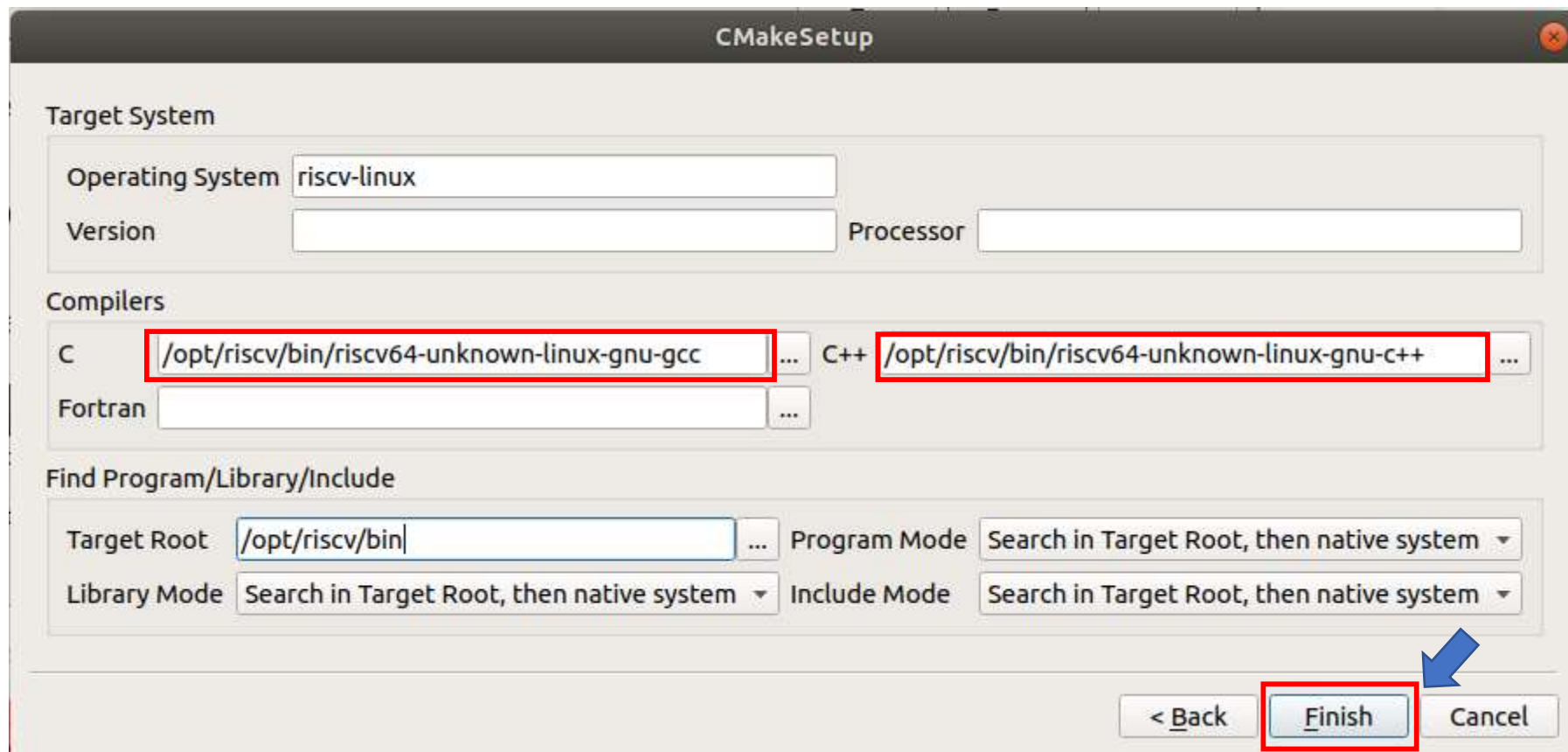
```
#include <stdio.h>  
  
int main()  
{  
    /* 我的第一个 C 程序 */  
    printf("Hello, World! \n");  
  
    return 0;  
}
```

交叉编译工具链验证过程

使用CMAKE对OpenCV进行编译



03 编译及运行

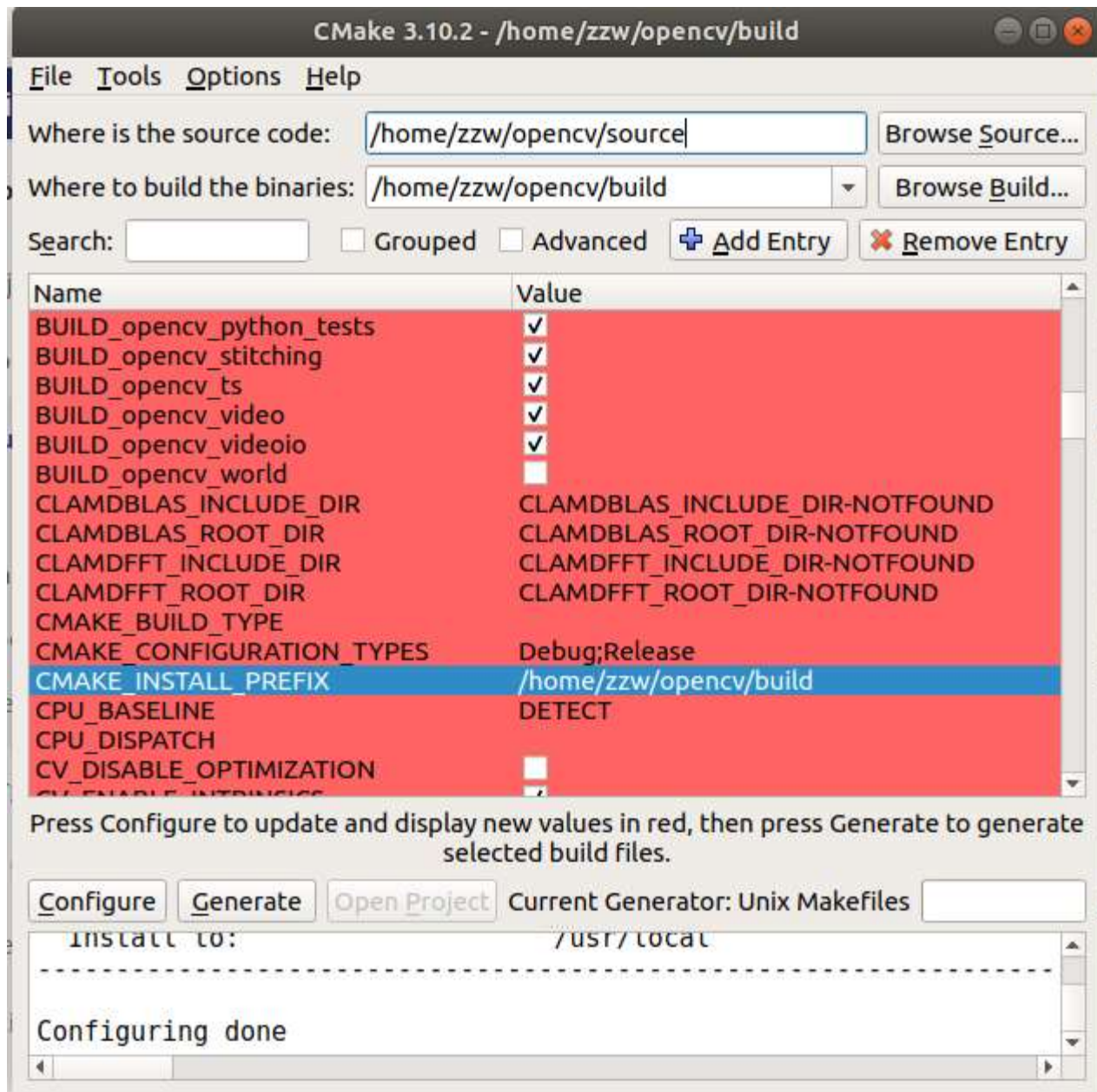


The image shows the CMakeSetup dialog box with the following fields and settings:

- Target System**
 - Operating System: riscv-linux
 - Version: (empty)
 - Processor: (empty)
- Compilers**
 - C: /opt/riscv/bin/riscv64-unknown-linux-gnu-gcc
 - C++: /opt/riscv/bin/riscv64-unknown-linux-gnu-c++
 - Fortran: (empty)
- Find Program/Library/Include**
 - Target Root: /opt/riscv/bin
 - Program Mode: Search in Target Root, then native system
 - Library Mode: Search in Target Root, then native system
 - Include Mode: Search in Target Root, then native system

At the bottom right, there are three buttons: < Back, Finish, and Cancel. The Finish button is highlighted with a red box, and a blue arrow points to it from the right.

03 编译及运行



03 编译及运行

The screenshot shows the CMake 3.10.2 GUI for the project located at `/home/zzw/opencv/build`. The source code is at `/home/zzw/opencv/source`. The configuration options are listed in a table:

Name	Value
BUILD_opencv_python_tests	✓
BUILD_opencv_stitching	✓
BUILD_opencv_ts	✓
BUILD_opencv_video	✓
BUILD_opencv_videoio	✓
BUILD_opencv_world	<input type="checkbox"/>
CLAMDBLAS_INCLUDE_DIR	CLAMDBLAS_INCLUDE_DIR
CLAMDBLAS_ROOT_DIR	CLAMDBLAS_ROOT_DIR
CLAMDFFT_INCLUDE_DIR	CLAMDFFT_INCLUDE_DIR
CLAMDFFT_ROOT_DIR	CLAMDFFT_ROOT_DIR
CMAKE_BUILD_TYPE	Debug;Release
CMAKE_CONFIGURATION_TYPES	Debug;Release
CMAKE_INSTALL_PREFIX	/home/zzw/opencv/build
CPU_BASELINE	DETECT
CPU_DISPATCH	
CV_DISABLE_OPTIMIZATION	<input type="checkbox"/>

Below the table, there is a message: "Press Configure to update and display new values in red, then press Generate to generate selected build files."

The "Configure" button is highlighted, and the "Current Generator" is set to "Unix Makefiles".

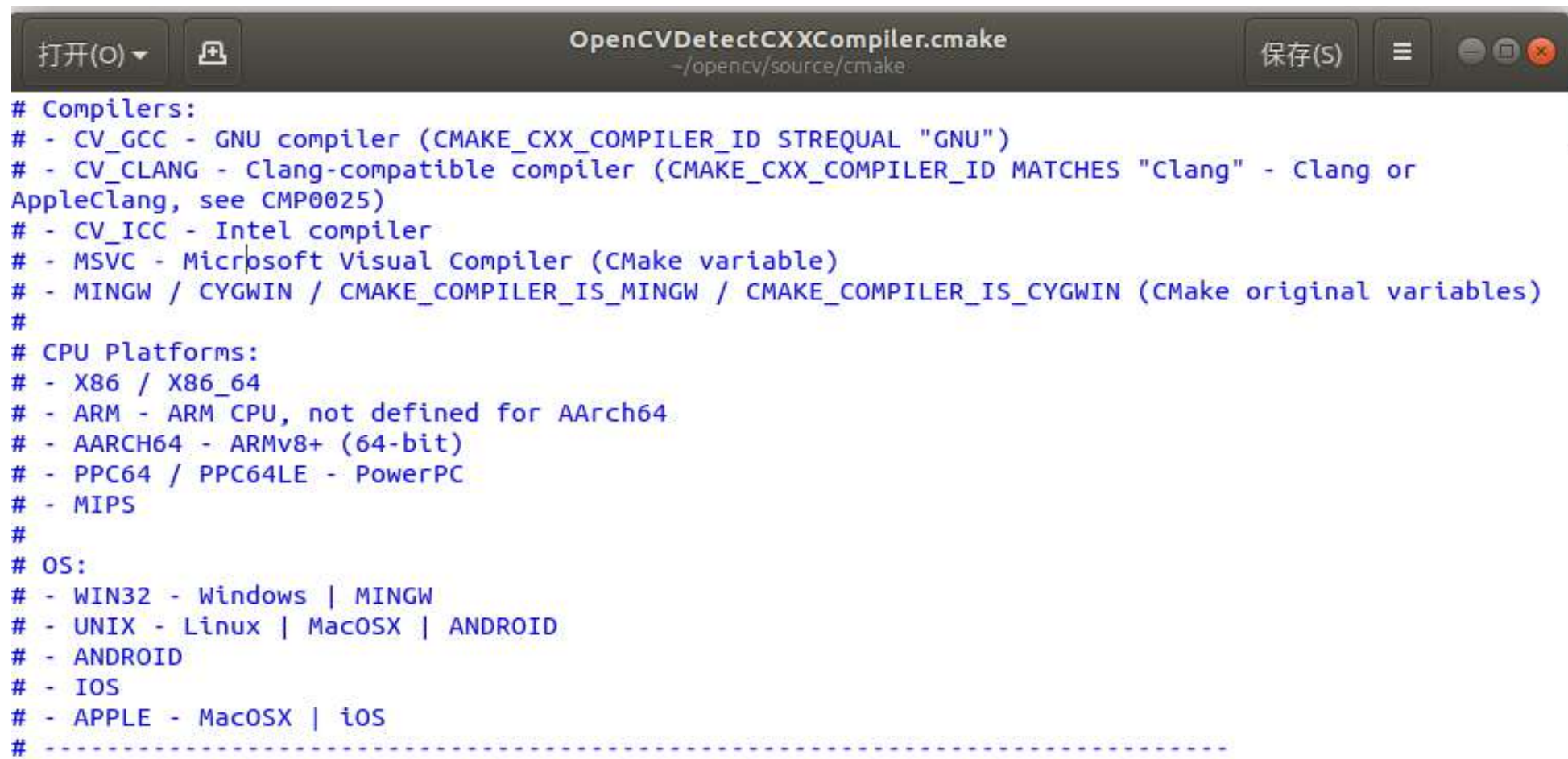
The output window shows the following messages:

```
Detecting C compile features - done
OpenCV: system-specific configuration file is not found: 'Generic'
CMake Warning at cmake/OpenCVDetectCXXCompiler.cmake:77 (message):
  OpenCV: CMAKE_SYSTEM_PROCESSOR is not defined. Perhaps CMake toolchain is
  broken
Call Stack (most recent call first):
  CMakeLists.txt:153 (include)
```

A blue arrow points from the text "问题说明：" to the warning message in the output window.

问题说明：
CMAKE_SYSTEM_PROCESSOR is not defined

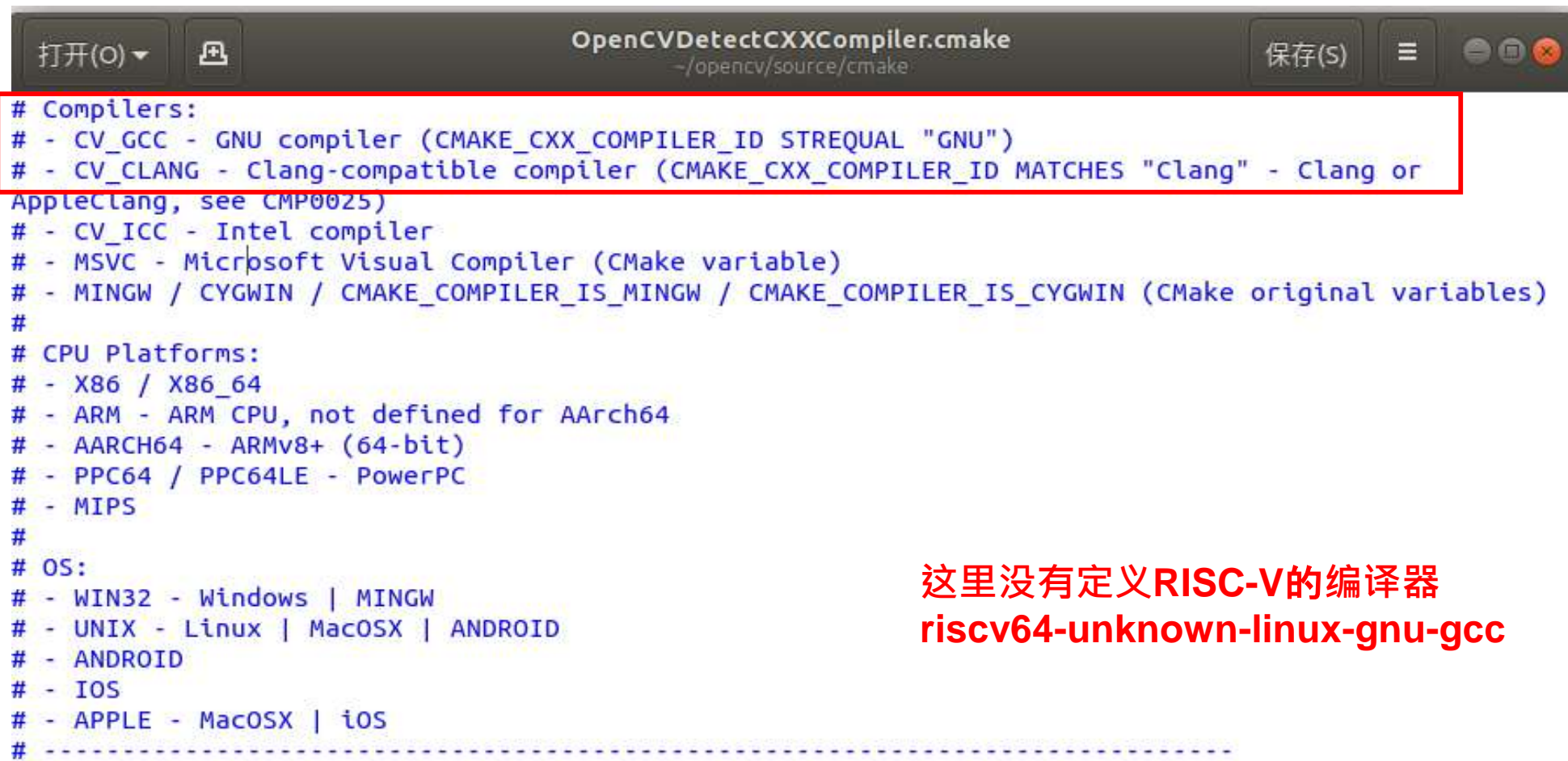
按照提示找到文件`cmake/OpenCVDetectCXXCompiler.cmake`，这是OpenCV定义编译的文件。



```
# Compilers:
# - CV_GCC - GNU compiler (CMAKE_CXX_COMPILER_ID STREQUAL "GNU")
# - CV_CLANG - Clang-compatible compiler (CMAKE_CXX_COMPILER_ID MATCHES "Clang" - Clang or
AppleClang, see CMP0025)
# - CV_ICC - Intel compiler
# - MSVC - Microsoft Visual Compiler (CMake variable)
# - MINGW / CYGWIN / CMAKE_COMPILER_IS_MINGW / CMAKE_COMPILER_IS_CYGWIN (CMake original variables)
#
# CPU Platforms:
# - X86 / X86_64
# - ARM - ARM CPU, not defined for AArch64
# - AARCH64 - ARMv8+ (64-bit)
# - PPC64 / PPC64LE - PowerPC
# - MIPS
#
# OS:
# - WIN32 - Windows | MINGW
# - UNIX - Linux | MacOSX | ANDROID
# - ANDROID
# - IOS
# - APPLE - MacOSX | iOS
# -----
```

03 编译及运行

按照提示找到文件`cmake/OpenCVDetectCXXCompiler.cmake`，这是OpenCV定义编译的文件。



```
# Compilers:
# - CV_GCC - GNU compiler (CMAKE_CXX_COMPILER_ID STREQUAL "GNU")
# - CV_CLANG - Clang-compatible compiler (CMAKE_CXX_COMPILER_ID MATCHES "Clang" - Clang or
AppleClang, see CMP0025)
# - CV_ICC - Intel compiler
# - MSVC - Microsoft Visual Compiler (CMake variable)
# - MINGW / CYGWIN / CMAKE_COMPILER_IS_MINGW / CMAKE_COMPILER_IS_CYGWIN (CMake original variables)
#
# CPU Platforms:
# - X86 / X86_64
# - ARM - ARM CPU, not defined for AArch64
# - AARCH64 - ARMv8+ (64-bit)
# - PPC64 / PPC64LE - PowerPC
# - MIPS
#
# OS:
# - WIN32 - Windows | MINGW
# - UNIX - Linux | MacOSX | ANDROID
# - ANDROID
# - IOS
# - APPLE - MacOSX | iOS
# -----
```

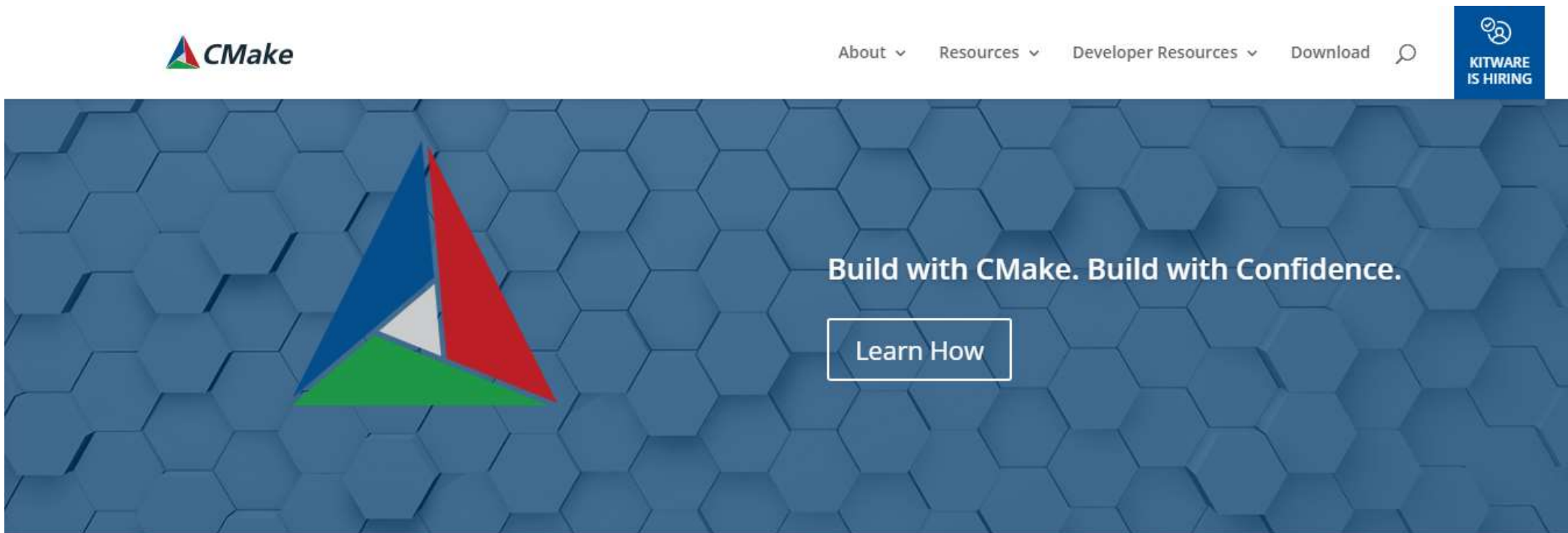
这里没有定义RISC-V的编译器
`riscv64-unknown-linux-gnu-gcc`

CMake是一个**跨平台**的安装（编译）工具，可以用简单的语句来描述所有平台的安装(编译过程)。它能够输出各种各样的makefile或者project文件，能测试编译器所支持的C++特性。

- CMake 可以编译源代码、制作程序库，还可以用任意的顺序建构执行档。
- CMake 支持 in-place 建构和 out-of-place 建构，因此可以很容易从同一个源代码目录树中建构出多个二进档。
- CMake 也支持静态与动态程式库的建构。

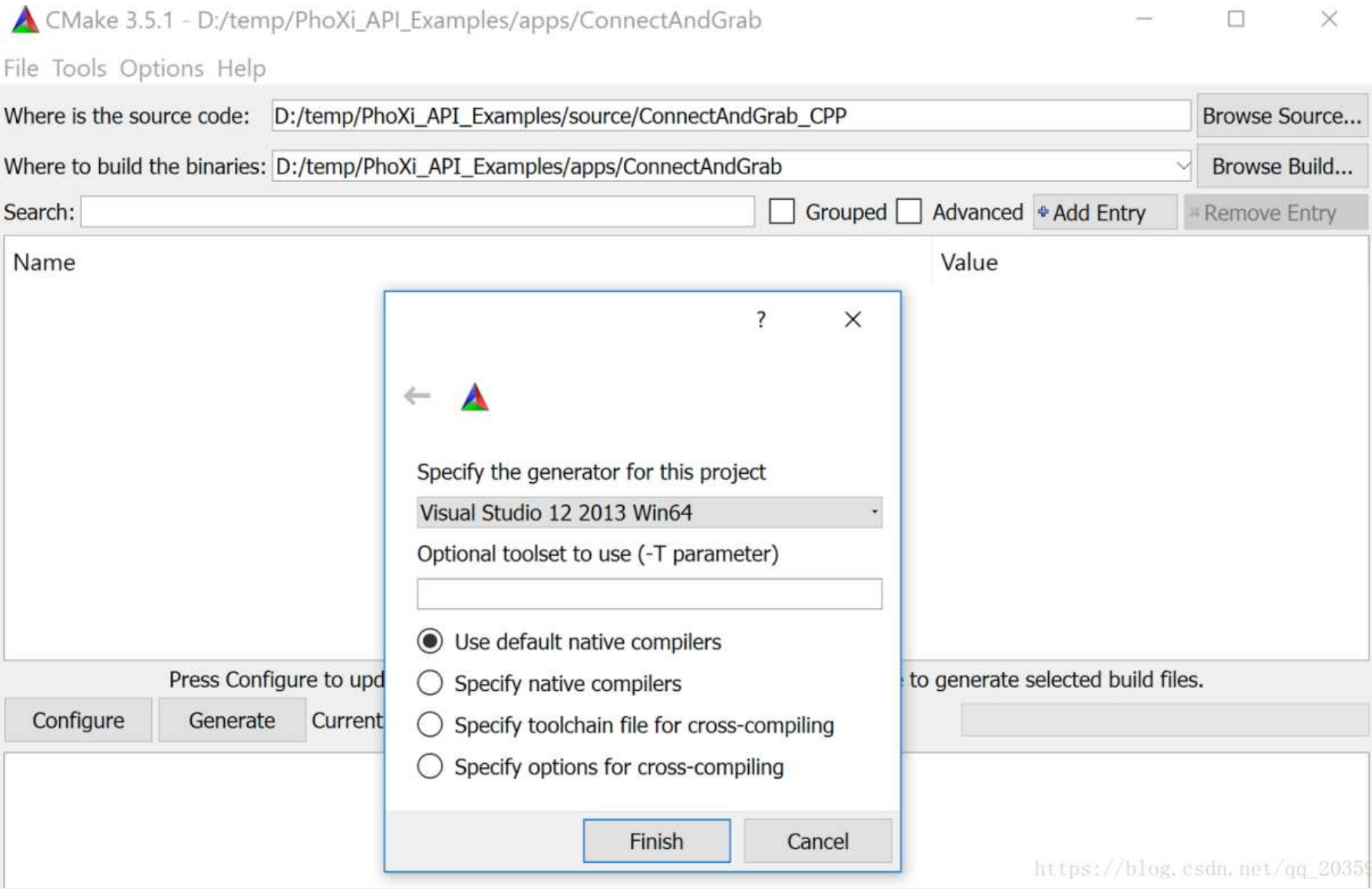
04 CMake系统介绍

CMAKE官网 <https://cmake.org/>



CAMKE官网截图

04 CMake系统介绍



CAMKE在WIN10下的运行界面

CMake vs Make

用于编写CMakeLists.txt文件的语言是可读的，也更容易理解。

它支持多个生成器，如Xcode、Eclipse、Visual Studio等。

系统库的跨平台发现。

以平台无关的方式将文件编译到共享库更容易，而且通常比make更容易使用。

的自动发现和配置工具链。

Linux下CMake的安装及简单使用

首先`cmake -version`查看下系统安装了没有

```
$ cmake -version
```

```
The program 'cmake' is currently not installed. You can install it by typing:  
sudo apt-get install cmake
```

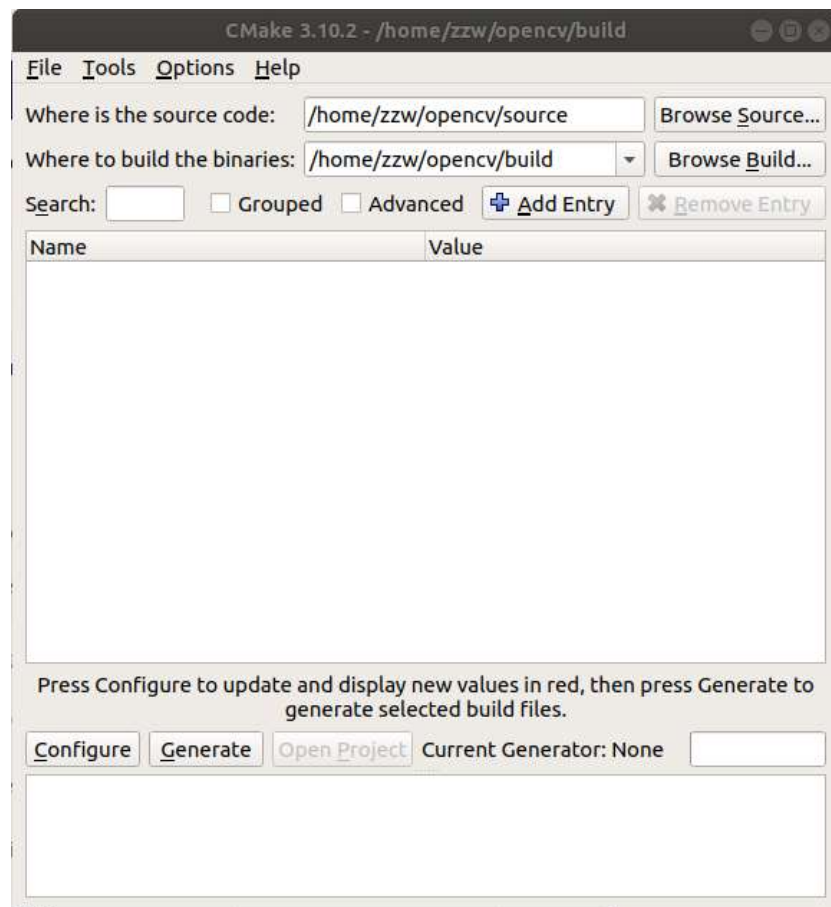
没有的话安装即可

```
$ sudo apt-get install cmake
```


04 CMake系统介绍

打开命令

```
$ sudo apt install cmake-qt-gui
```



CMakeLists.txt

即cmake所处理的“代码”。其实，使用cmake管理项目本身也是在编程，所以称之为“代码（或脚本）”并不为过。

```
#cmake最低版本需求，不加入此行会受到警告信息  
CMAKE_MINIMUM_REQUIRED(VERSION 2.6)  
PROJECT(HELLO) #项目名称  
#把当前目录(.)下所有源代码文件和头文件加入变量SRC_LIST  
AUX_SOURCE_DIRECTORY(. SRC_LIST)  
#生成应用程序 hello (在windows下会自动生成hello.exe)  
ADD_EXECUTABLE(hello ${SRC_LIST})
```

cmake检测编译环境，并生成相应的**makefile**。接着，运行命令make进行编译。编译后，生成的所有中间文件和可执行文件会在build目录下。

谢 谢

欢迎交流合作

2019/02/5