# Berry Pi : QT with OpenCV to use TIS camera

## → Install opency

### 1. Update the package list and upgrade:

- I. sudo apt-get update
- II. iisudo apt-get upgrade
- III. sudo apt-get dist-upgrade -y

#### 2. Build and compile the required tools:

sudo apt-get install build-essential gcc cmake pkg-config

#### 3. Intsll Git:

sudo apt-get install git

#### 4. Graphics window library:

sudo apt-get install libgtk2.0-dev

**5.** Graphic audio video codec, recording, conversion, streaming: sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev

#### 6. Image format::

sudo apt-get install libjpeg-dev libpng-dev libtiff-dev libjasper-dev

sudo apt-get install -y libtbb-dev libeigen3-dev

## 7. Download and build OpenCV:

Download the OpenCV source code in GitHub with git and get the directory opency:

I. git clone https://github.com/Itseez/opencv.git

II. cd opencv

The current release of the official version of 3.2.0, I was 3.1.0 to do the model:

III. git checkout 3.1.0

IV. mkdir build

V. cd build

Execute cmake to generate the desired profile for building:

□ -D CMAKE\_INSTALL\_PREFIX=/usr/local □ On behalf of the path to be installed, the last "..." represents the source code where the path.

cmake -D CMAKE\_BUILD\_TYPE=Release -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

Began to build (due to Berry PI cpu speed is very slow, about 10 hours to build, so add the parameter "-j4" to quad-core speed, but still about 3 hours

make -j4

install :

VI. sudo make install

Execute the instruction update library :

VII. sudo ldconfig

Check the version of OpenCV

VIII. pkg-config --modversion opencv Result: 3.1.0

## **二** ヽ Install QT

I. sudo apt-get install qt5-default

II. sudo apt-get install qtcreator

# $\equiv$ **、 QT** with **OpenCV** to use **TIS** camera

## I. Select the file name and file location

Introduction and Project Location Kits Details Summary Introduction and includes an empty widget. Name: opencamera Create in [/home/pi Use or default project location		Qt Widgets Application	-		×
Kits       This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.         Summary       Name:         Opencamera       Create in [/home/pi         Browse       Browse		Introduction and Project Location			
Name: opencamera Create in /home/pi Browse Browse	Kits Details Summary	This wizard generates a Qt Widgets Application project. The application derives by default fror QApplication and includes an empty widget.	n		
TH- Create in /home/pi Browse		Name: opencamera			
	ТН	Create in //home/pi	owse		
		Next >	Can	cel	

## **II. Kit Selection**

s	Ot Creator can use the following kits for project opencamera:	
tails mmary	🖉 🔙 Desktop	Deta

# III. Build and Run Compliers select

	Options	- 0
Filter	Build & Run	
Environment	General Kits Qt Versions Compilers Debuggers CMake	
Text Editor	Name Type	Add
🖌 FakeVim	Auto-detected	
2) Help	GCC GCC	Bemove
C++	ABI: not up-to-date	
S Build & Run		
Debugger	Name: GCC	
🗶 Designer	Compiler nath:	nwse
Analyzer		
Yersion Control	Platform codegen flags:	
💠 Android	Platform linker flags:	
BlackBerry	ABI: unknowr + - unknowr + - unknowr + - unknowr + - unknowr unknowr	nowr -
ALL ONLY		

Kits -> Complier -> GCC

		Options		_ = >
Filter	Build & Run			
Environment	General Kits Q	t Versions Compilers Debuggers CMake		
Text Editor	Name			Add
😸 FakeVim	Auto-detected			Clone
С++	Desktop	(default)		Remove
🖉 Qt Quick				Make Default
🕵 Build & Run	File system nan	ne:		^
Debugger	Device type:	Desktop		
🗶 Designer	Device:	Local PC (default for Desktop)	•	Manage
Analyzer	Sysroot: 🖊			Browse
Version Control	Compiler:	GCC	•	Manage
Android	Debugger:	System GDB at /usr/bin/gdb	•	Manage
BIACKBERRY	Qt version:	Qt 5.3.2 (qt5)		Manage
v v			Apply	cel OK

## **IV.** Introducing the library

a. You can use the terminal to find the location of include and lib

pkg-config --cflags opencv

pkg-conf -libs opencv

pi@raspberrypi:~ \$ pkg-configcflags opencv
-I/usr/local/include/opencv -I/usr/local/include
pi@raspberrypi:~ \$ pkg-configlibs opencv
-L/usr/local/lib -lopencv_shape -lopencv_stitching -lopencv_objdetect -lopencv_superres -lopencv_videos
tab -lopencv_calib3d -lopencv_features2d -lopencv_highgui -lopencv_videoio -lopencv_imgcodecs -lopencv_
video -lopencv_photo -lopencv_ml -lopencv_imgproc -lopencv_flann -lopencv_core

- b. open .pro
- c. Enter :

```
INCLUDEPATH += /usr/local/include/
```

LIBS += -L/usr/local/lib -lopencv\_highgui -lopencv\_imgcodecs -lopencv\_imgproc lopencv\_core

As the imaging source usb2 camera in the Linux version for the bayer mode output, and OpenCV in the library VideoCapture and cvcapture can only accept the RGB image output format, resulting in can not directly use these two libraries to open the camera, so we Must be directly to v412 for processing and then displayed.

# The following is a converted sample program connection URL, the user can quickly open the camera through the following steps to do the application: https://gist.github.com/TIS-Edgar/10f04501f49b6b3bf75e

#### 1. Download ZIP

GitHubGist Search					All gists GitHub				New gist		
TIS-Edgar / N Created 3 years ago	lakefile						\star Star	1	¥ For	k 1	(!) •
<>Code -o- Rev	visions 1	🔺 Stars 1	🎾 Forks 1		Embed 🔻	<script src="https:</th> <th>//gist.</th> <th></th> <th>ţ</th> <th>Downloa</th> <th>ad ZIP</th>	//gist.		ţ	Downloa	ad ZIP

- 2. Unzip the file and put main.cpp, v4ldevice.cpp, v4ldevice.h into your QT project folder
- 3. Right-click in the project and select "add Existing Files"



4. Add <u>v4ldevice.cpp</u> v4ldevice.h main.cpp

