Berry Pi : QT with OpenCV to use TIS camera

一、 Install opencv

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 opencv:

```
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

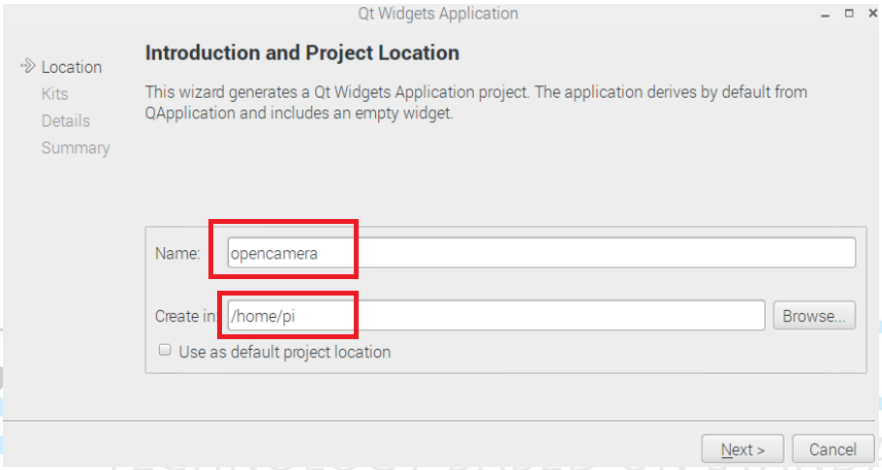
二、 Install QT

I. sudo apt-get install qt5-default

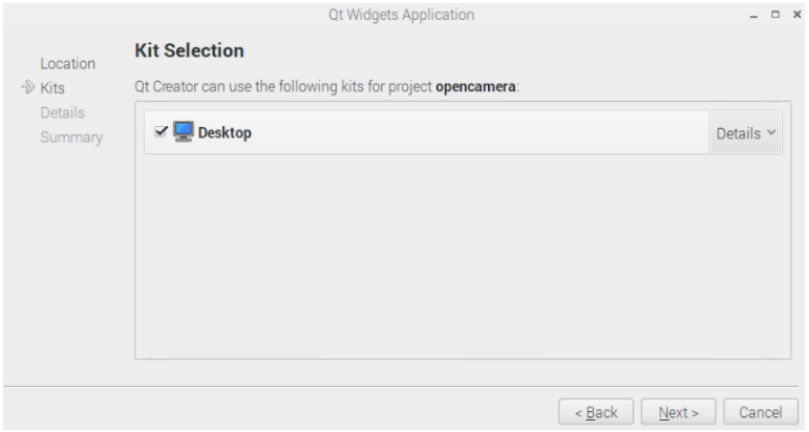
II. sudo apt-get install qtcreator

三、 QT with OpenCV to use TIS camera

I. Select the file name and file location

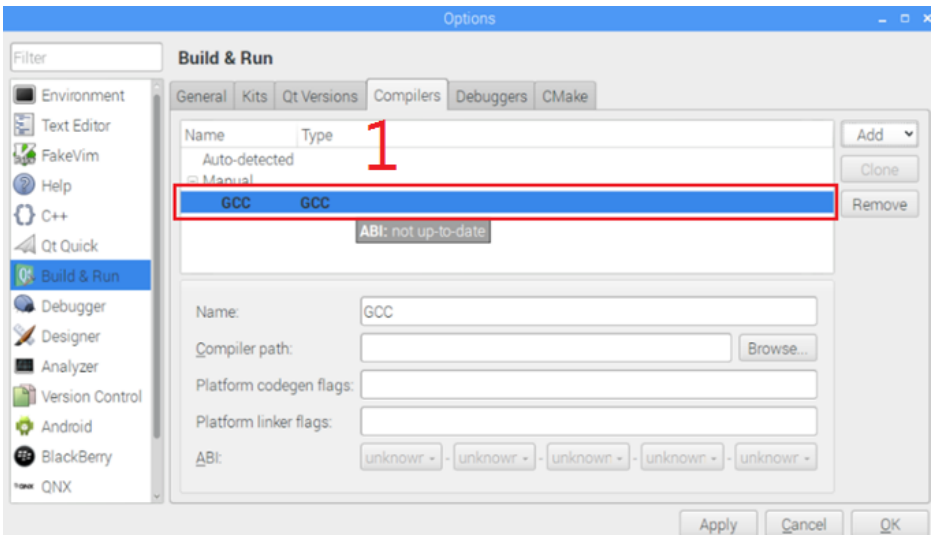


II. Kit Selection

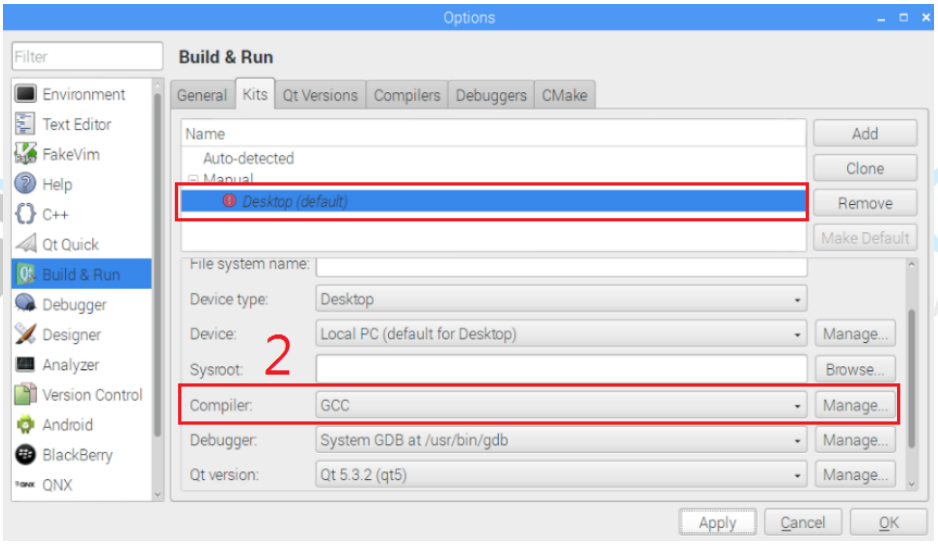


III. Build and Run Compliers select

Tools -> Optins -> Build & Run -> Compliers ->Add->GCC



Kits -> Complier -> GCC



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-config --cflags opencv
-I/usr/local/include/opencv -I/usr/local/include
pi@raspberrypi:~ $ pkg-config --libs 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 :

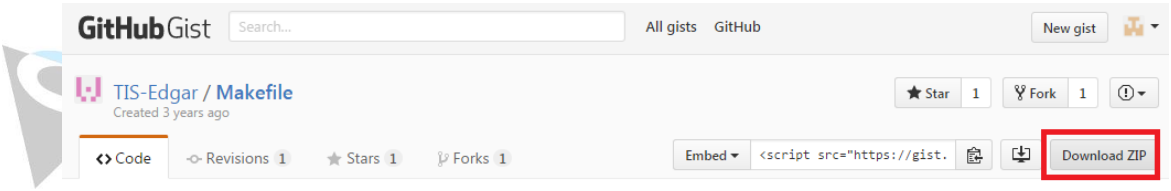
```
INCLUDEPATH += /usr/local/include/

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 v4l2 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:

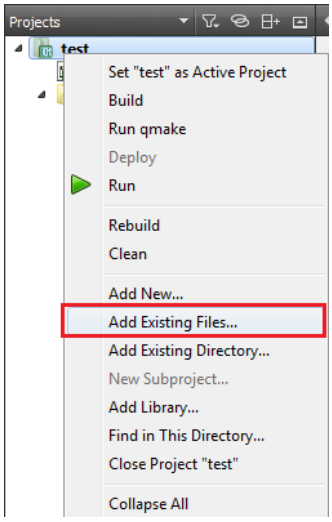
https://gist.github.com/TIS-Edgar/10f04501f49b6b3bf75e

1. **Download ZIP**



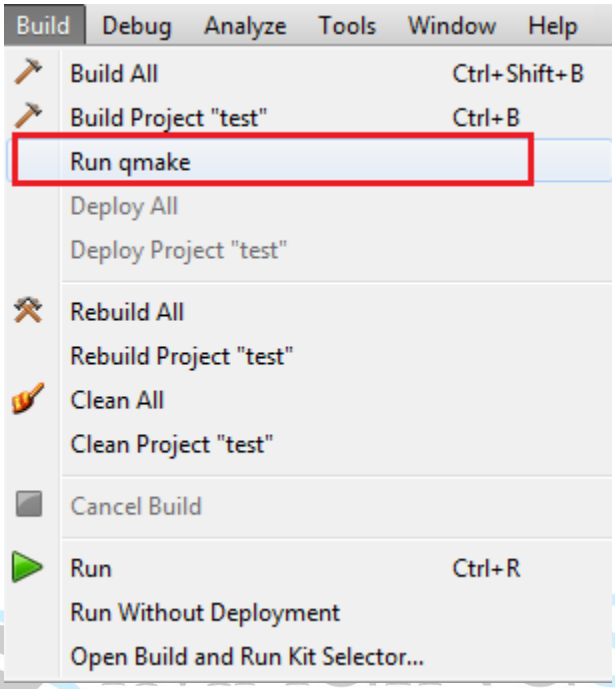
2. **Unzip the file and put `main.cpp`, `v4ldevice.cpp`, `v4ldevice.h` into your QT project folder**

3. **Right-click in the project and select "add Existing Files"**



4. **Add `v4ldevice.cpp`、`v4ldevice.h`、`main.cpp`**

5. **Build ->Run qmake**
6. **Build->Run**



7. **Result**