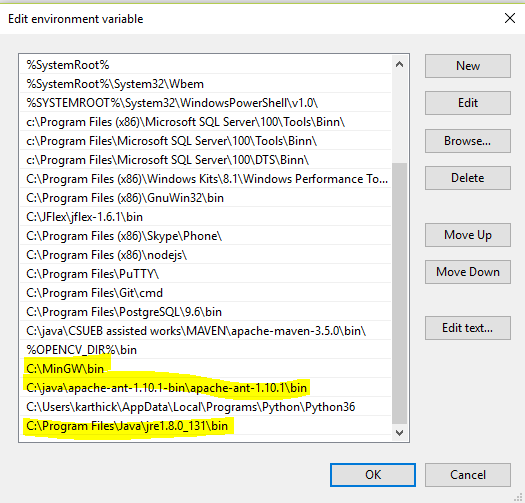
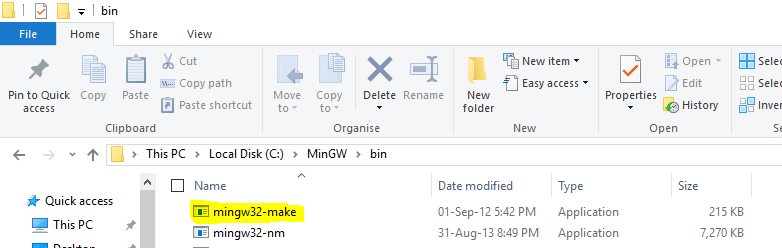
**Prerequisites:**

* **Ant**
* **OpenCV free and non free source code**
* **Java 8( JDK and JRE has to be same version)**
* **Android NDK**
* **Cmake**
* **MinGW**
* **Msys**

**Set the environment variables for MinGW, Ant, JDK, JRE on windows system environments**



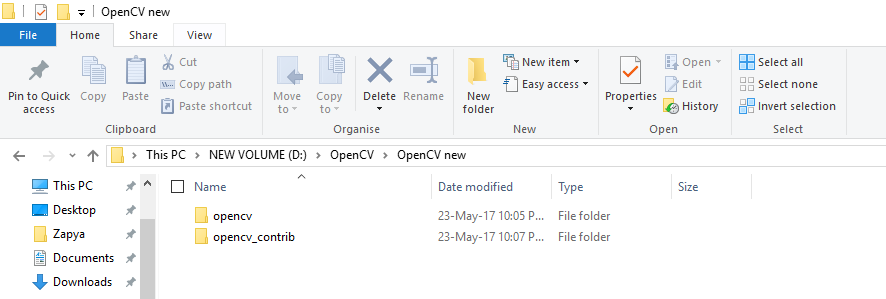
Rename the make to mingw32-make as shown below:



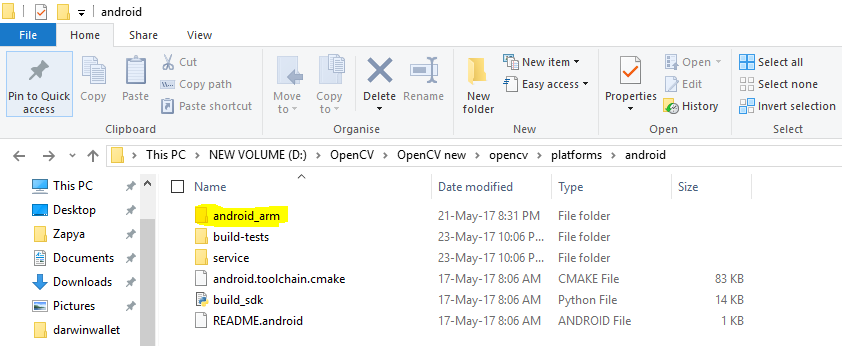
1. Download the OpenCV and OpenCV\_contrib source code from opencv github and place both the folder in a same directory.

OpenCV: <https://github.com/Itseez/opencv>

Extra Modules: <https://github.com/Itseez/opencv_contrib>



1. Create a folder android\_arm under opencv/platforms/android



Open opencv\modules\features2d\misc\java\src\cpp\ features2d\_manual.cpp

Add #include "opencv2/xfeatures2d.hpp"

Add the below line at line 121:

case SIFT:

fd = xfeatures2d::SIFT::create();

break;

case SURF:

fd = xfeatures2d::SURF::create();

break;

Add the below line at line 240:

case SIFT:

de = xfeatures2d::SIFT::create();

break;

case SURF:

de = xfeatures2d::SURF::create();

break;

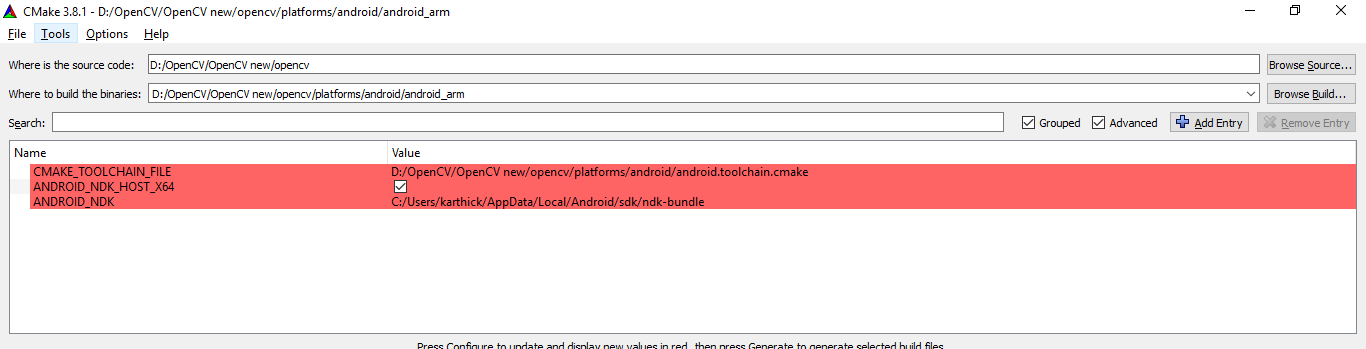
Save.

1. Open Cmake-GUI application

Select opencv directory as source code

Select opencv/platforms/android/android\_arm for “build the binaries”

* ANDROID\_NDK, type 'path', value '../ndk-dir'
* ANDROID\_NDK\_HOST\_X64, type 'bool', value 1
* CMAKE\_TOOLCHAIN\_FILE, type 'path', value '../opencv/platforms/android/android.toolchain.cmake'



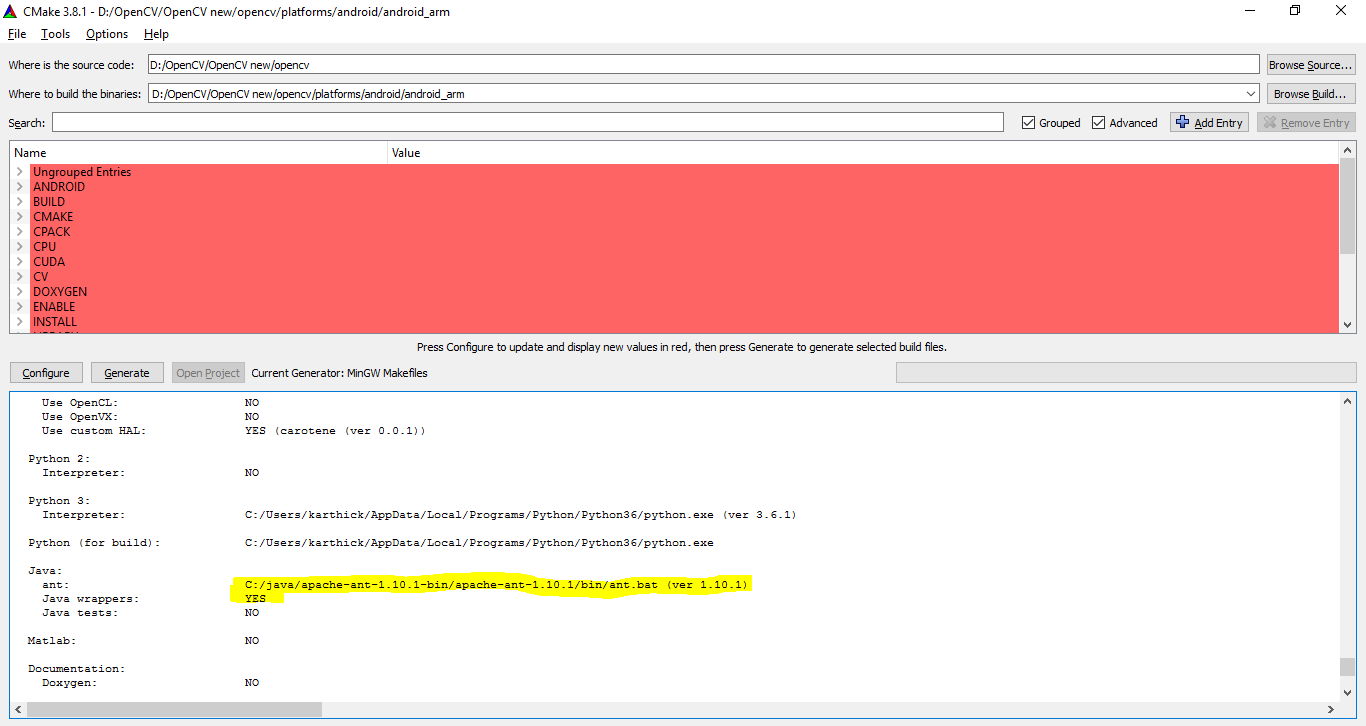
Press configure

Select MinGW Makefiles under “select the generator for this project”

Check the radiobutton “specify toolchain file for cross-compiling “

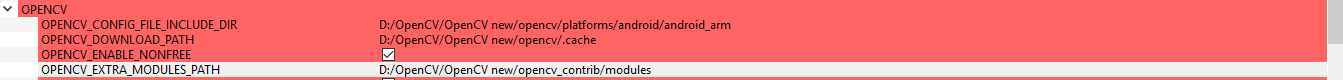
Click Next. Select '../opencv-source/platforms/android/android.toolchain.cmake' for toolchain compile option on next window. Click finish. This will start the configuration.

The log window will display the information. Check for any errors.

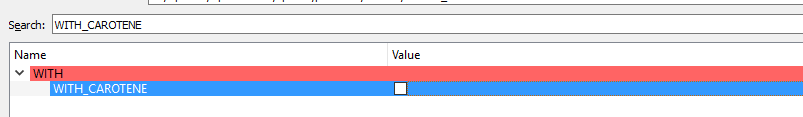


Make sure Ant directory is caught during configuration. Else it won’t generate Java files.

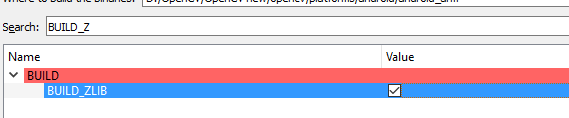
Update the extra modules configuration as below:



Uncheck WITH\_CAROTENE



Check BUILD\_ZLIB



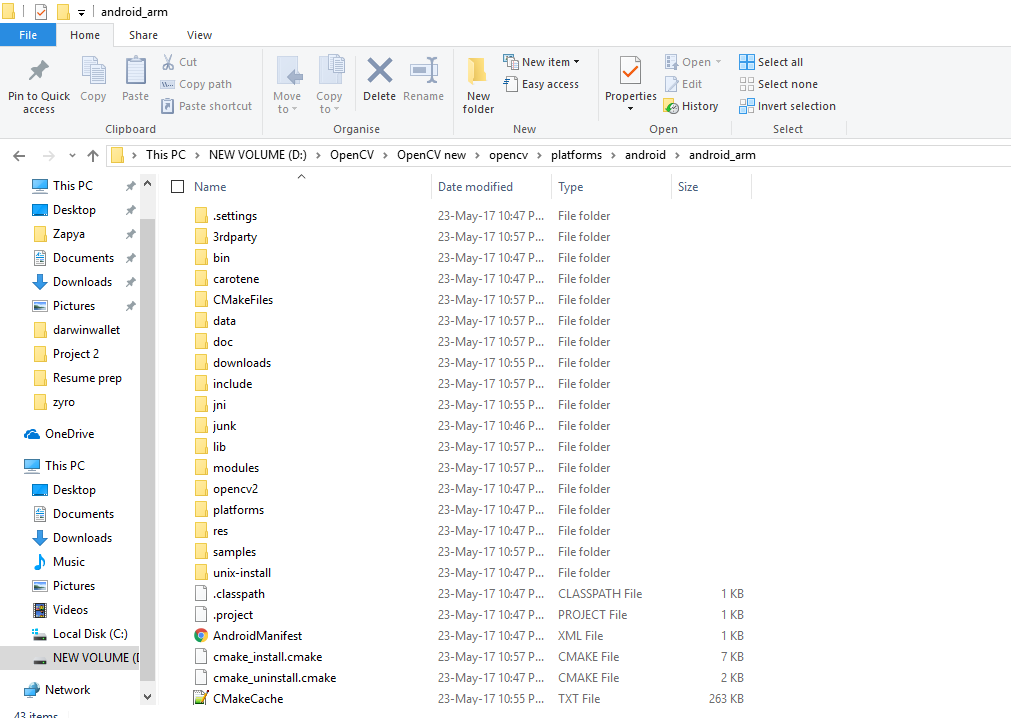
**Make sure this check box is on**



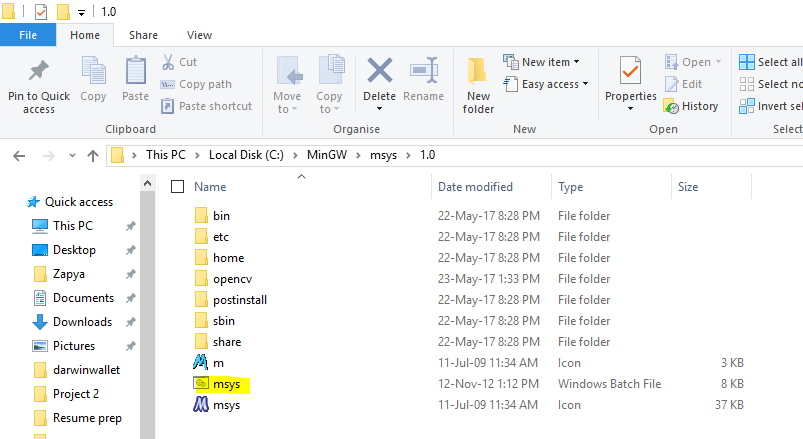
Press configure again

Click on Generate after completion

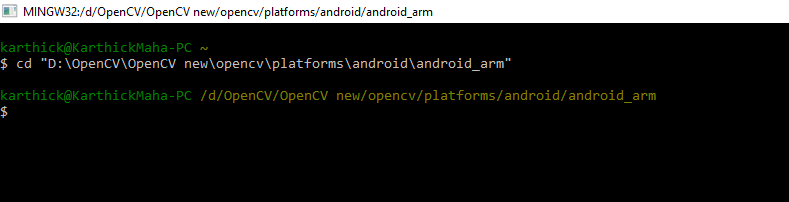
Once the Generation is complete, this will generate the files at target dir as shown below:



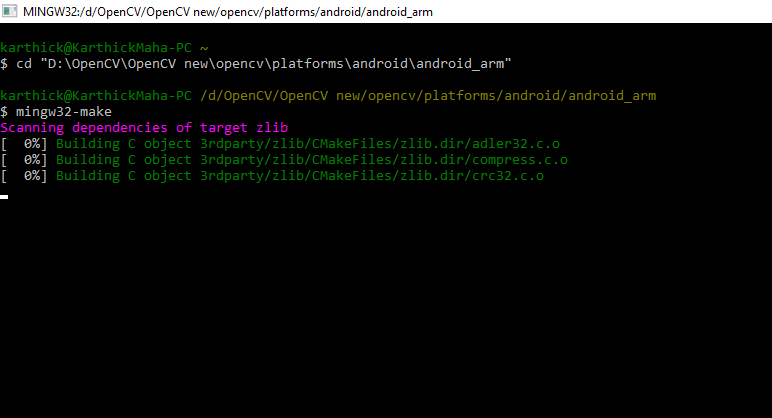
Go to '../mingw-dir/msys/1.0' and run 'msys' bash file



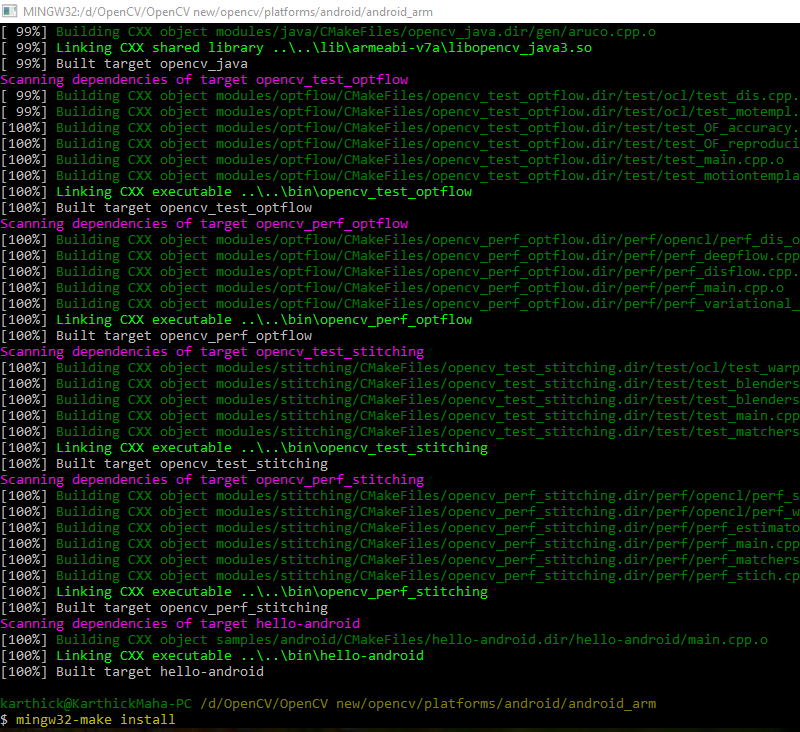
Change directory to the place where you generated make files.



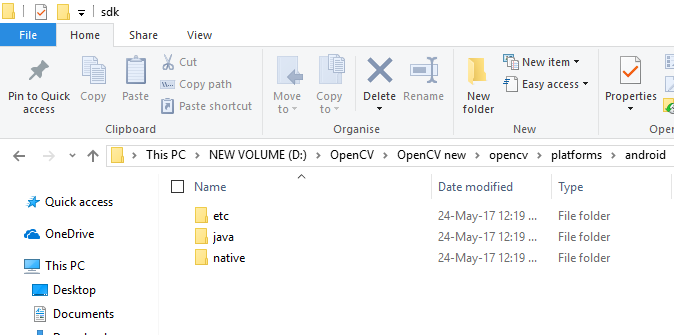
Run 'mingw32-make' command(This will take an hour to compile the entire codes)



Once completed type mingw32-make install



Post completion, you will find the sdk created with java and native lib support. Import the OpenCv under opencv\platforms\android\android\_arm\install\sdk\java for your android application.



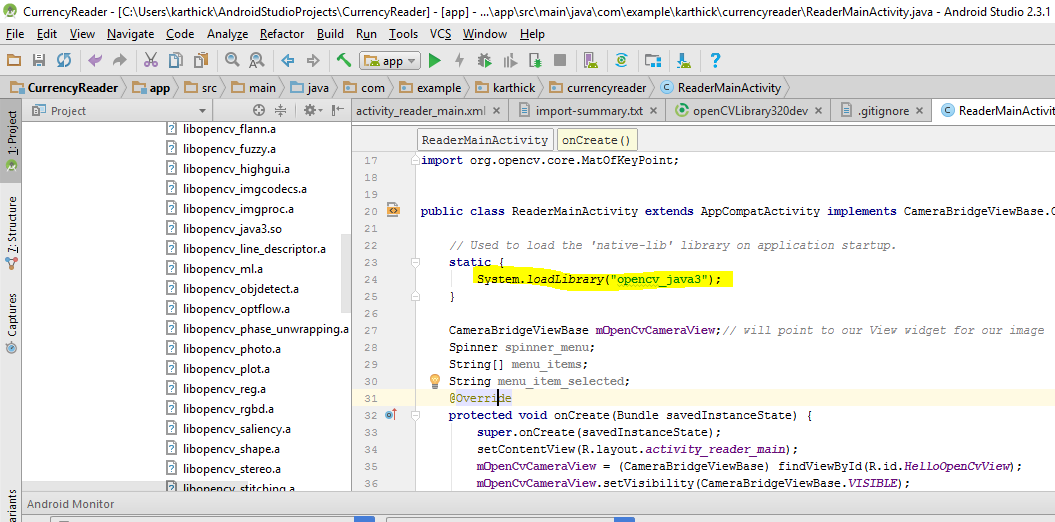
Create a new Android project. Check the Enable C++ support while creating the project.

After you create new project, copy the files under opencv\platforms\android\android\_arm\install\sdk\native\libs\armeabi-v7a to your

Project-Dir\app\src\main\jniLibs\armeabi-v7a

Update your java file(MainActivity.java) in your project to load the newly copied native code.

**static** {  
 System.*loadLibrary*(**"**opencv\_java3**"**);  
}



With this, you can use the features of SIFT and SURF

For further reference follow

<https://stackoverflow.com/questions/40948953/building-opencv-for-android-and-using-it-with-the-ndk>

I have compiled OpenCV extra modules that supports SIFT and SURF and uploaded in dropbox. Please find the link for reference.

<https://www.dropbox.com/sh/t444lnuxnd0knn7/AABza8sWDSJVC3Rh3_SfKv_ua?dl=0>

Sample project that demonstrates SIFT in OpenCV

<https://www.dropbox.com/sh/kh5dkv77bk9cixy/AADknlU8-qRCNuVn6hzez4zTa?dl=0>