INSTALACJA OPENCV v 3.2

Kody źródłowe i wersje release dostępne są pod adresem: http://opencv.org/releases.html

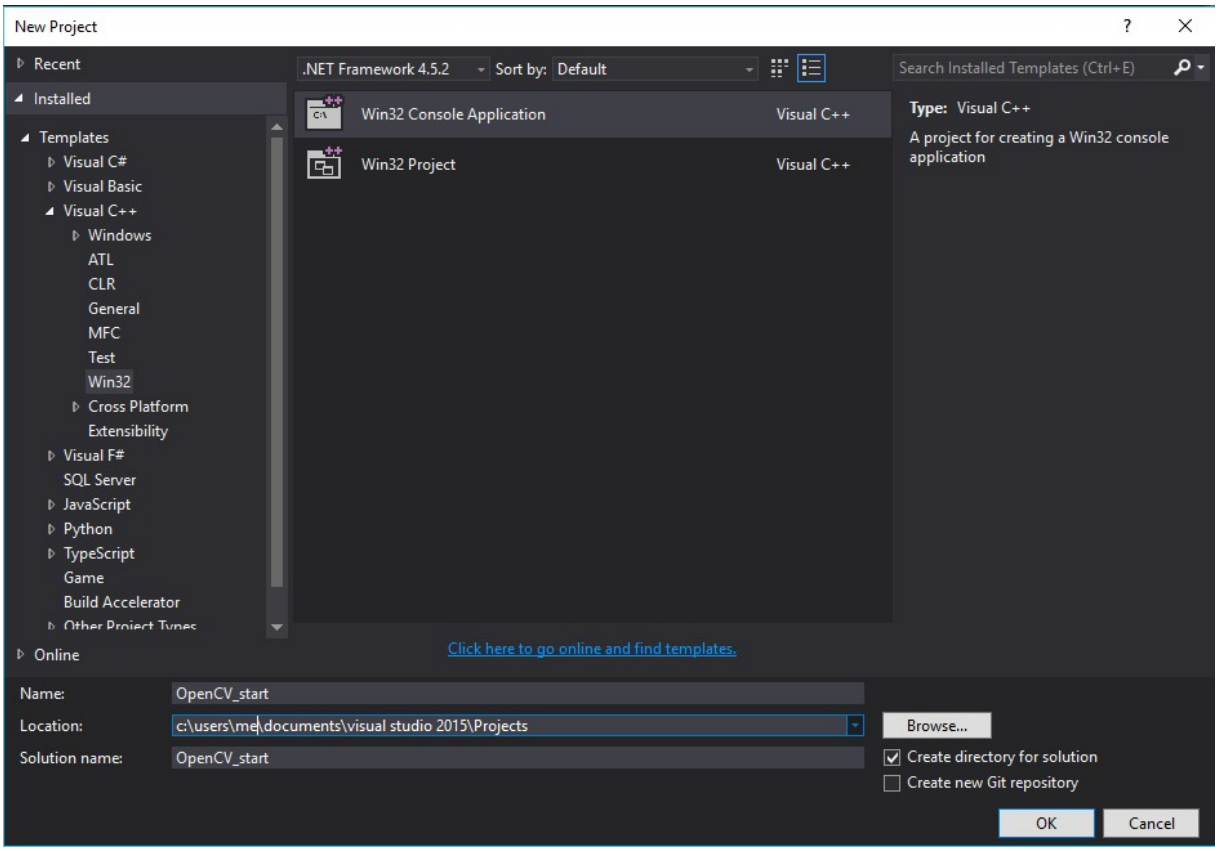
Windows

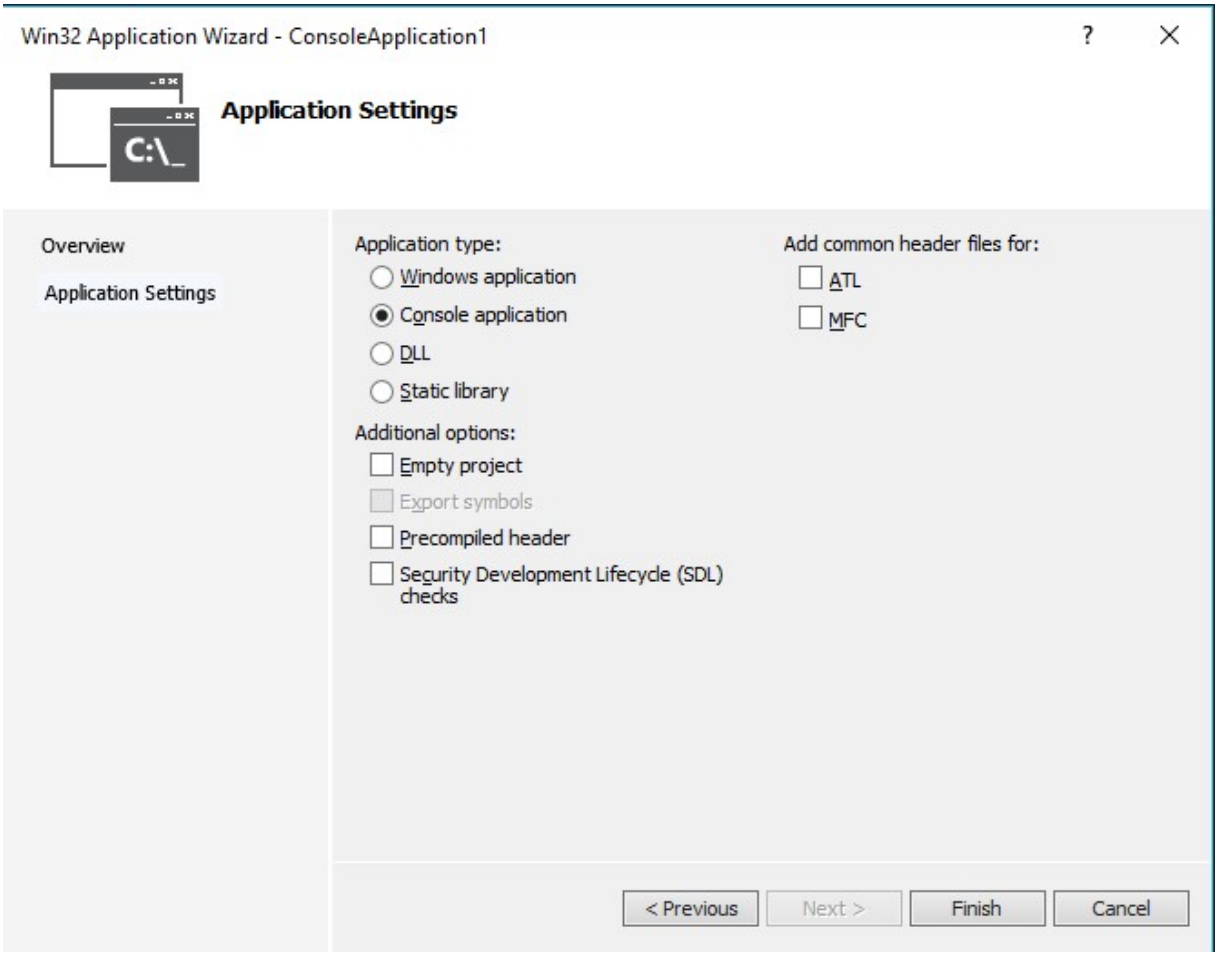
1. Skorzystanie z wersji release (binarnej)

    - Ze strony http://opencv.org/releases.html ściągnąć wersję Win pack (https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.2.0/opencv-3.2.0-vc14.exe/download) – biblioteki kompatybilne z Visual Studio od wersji 2015 oraz minGW, w wersji 64 bit
    - Rozpakować plik do dowolnego katalogu (np. c:\Opencv_3_2)
    - W wybranym IDE dodać katalog z bibliotekami jako c:\Opencv_3_2\opencv\build\x64\vc14\lib oraz katalog z plikami nagłówkowymi jako c:\Opencv_3_2\opencv\build\include
    - Dodać do projektu linkowanie z biblioteką opencv_world320d.lib (dla wersji debug) lub opencv_world320.lib (dla wersji release)
    - Pliki .dll potrzebne do uruchomienia znajdują się w katalogu c:\Opencv_3_2\opencv\build\x64\vc14\bin – należy je przekopiować w miejsce, w którym wybrane IDE generuje pliki wykonywalne

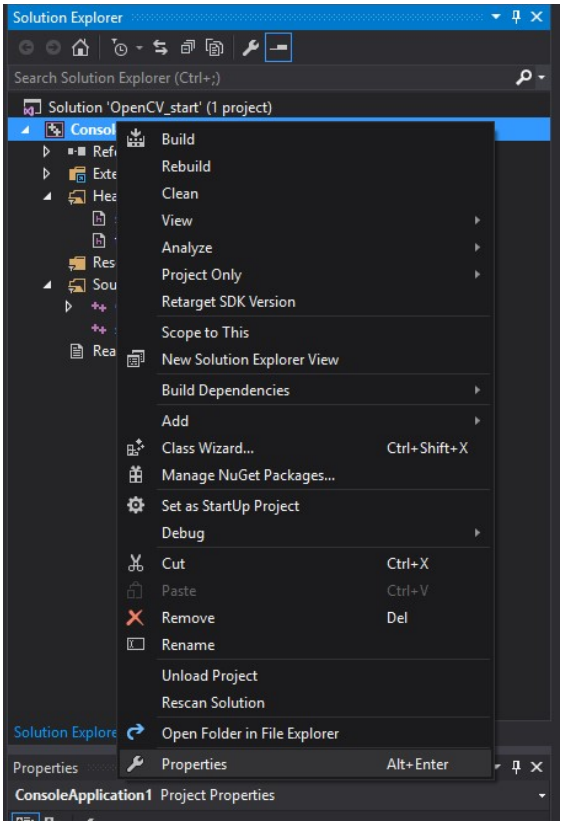
Konfiguracja na przykładzie Visual Studio:

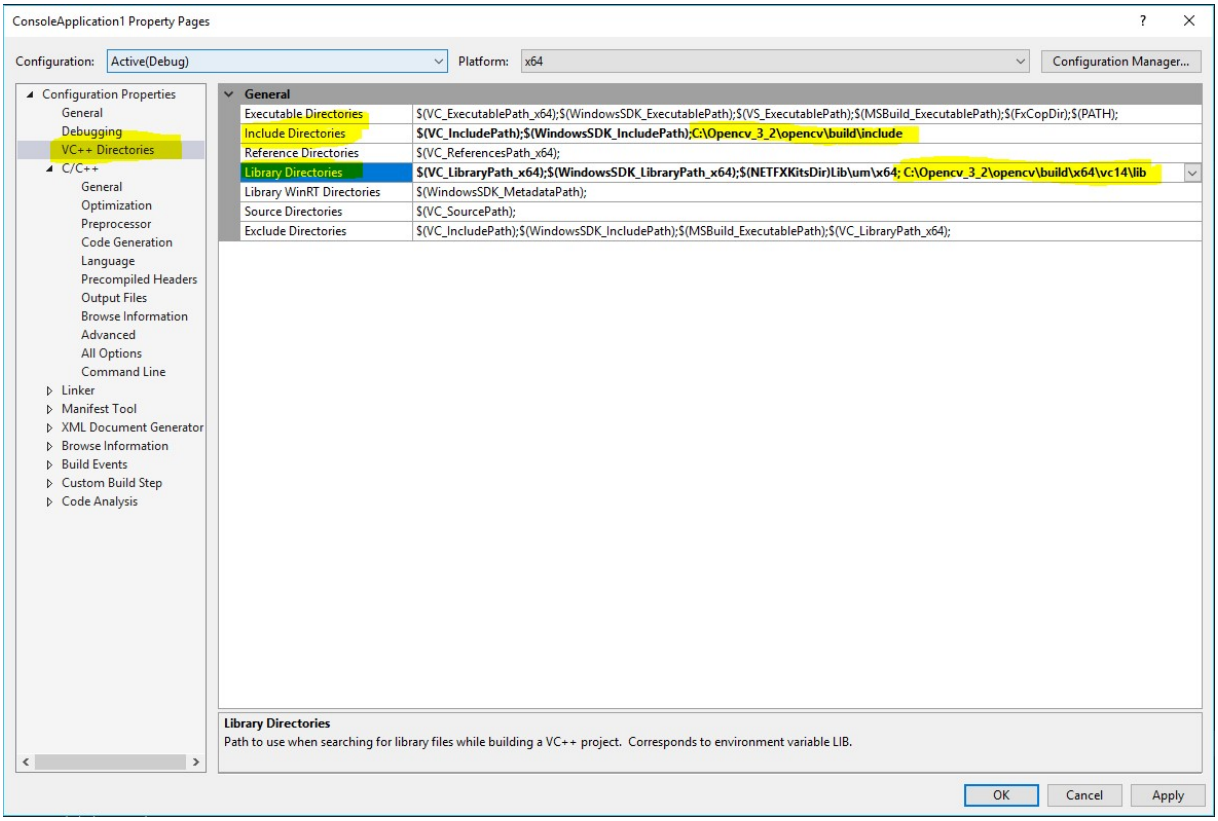
1. Utworzenie nowego projektu

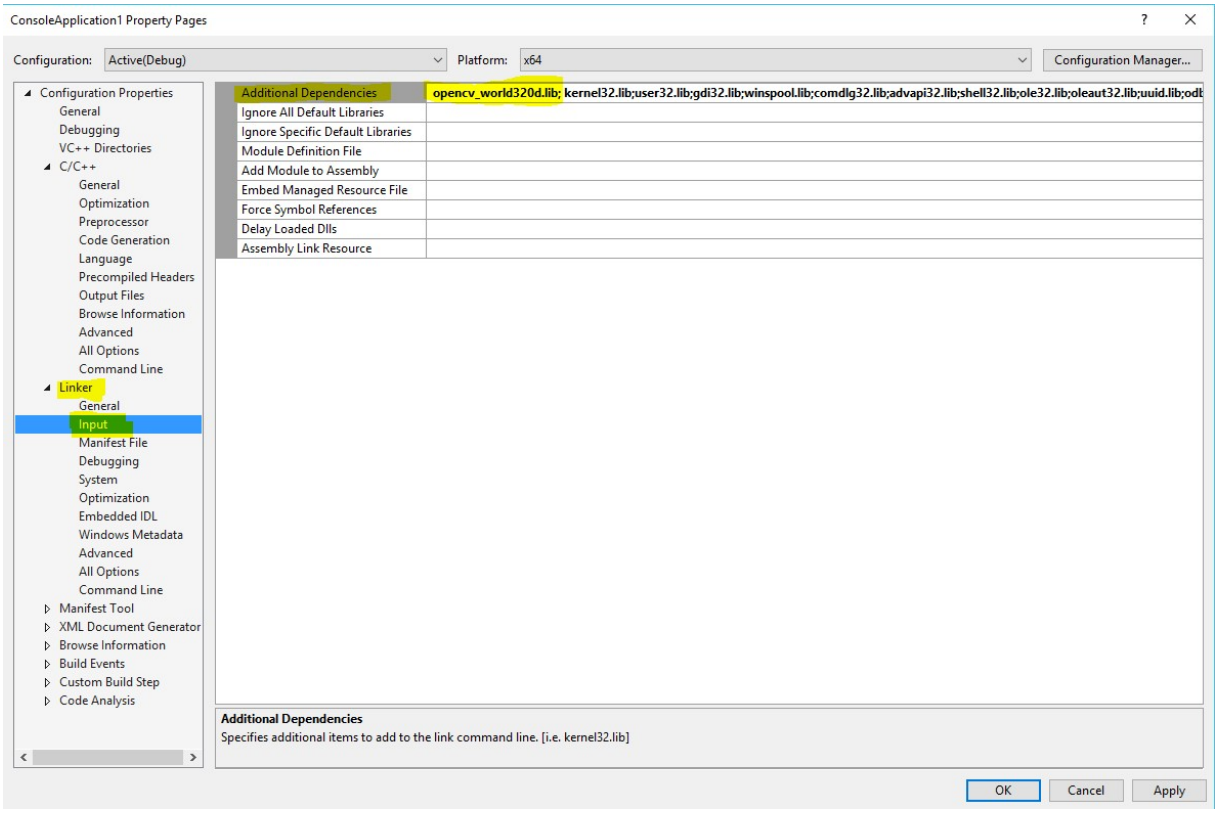

New Project
Recent
Installed
Templates
Visual C#
Visual Basic
Visual C++
Windows
ATL
CLR
General
MFC
Test
Win32
Cross Platform
Extensibility
Visual F#
SQL Server
JavaScript
Python
TypeScript
Game
Build Accelerator
Other Project Types
Online
.NET Framework 4.5.2
Sort by: Default
Win32 Console Application
Visual C++
Win32 Project
Visual C++
Search Installed Templates (Ctrl+E)
Type: Visual C++
A project for creating a Win32 console application
Click here to go online and find templates.
Name: OpenCV_start
Location: c:\users\me\documents\visual studio 2015\Projects
Solution name: OpenCV_start
Browse...
Create directory for solution
Create new Git repository
OK
Cancel


Win32 Application Wizard - ConsoleApplication1
Application Settings
Overview
Application Settings
Application type:
Windows application
Console application
DLL
Static library
Additional options:
Empty project
Export symbols
Precompiled header
Security Development Lifecycle (SDL) checks
Add common header files for:
ATL
MFC
< Previous
Next >
Finish
Cancel

2. Dodanie zależności



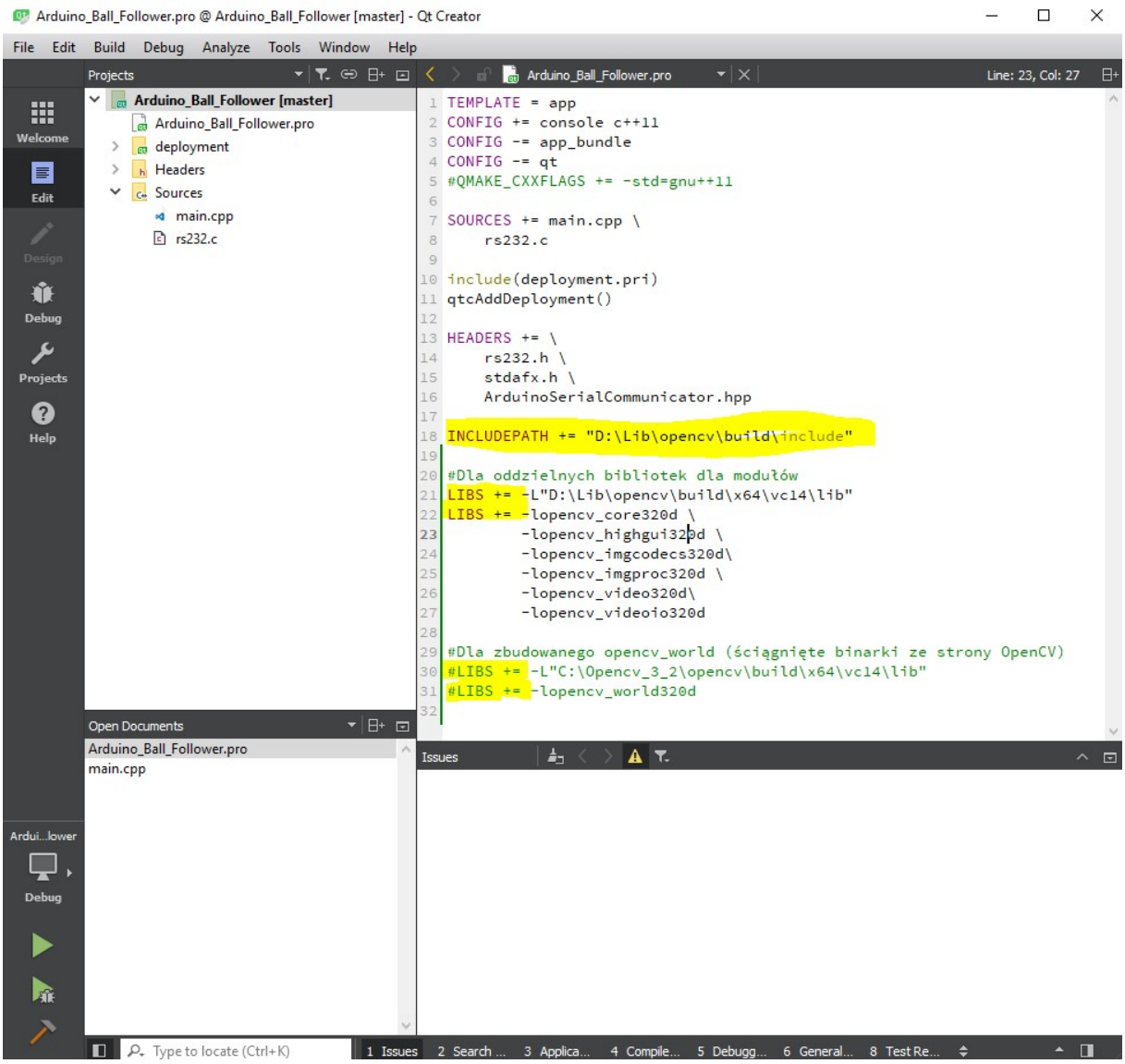




Konfiguracja na przykładzie QT Creator:

1. Dodanie bibliotek OpenCV do projektu:
   W pliku .pro (właściwości projektu dodać następujące wpisy (TYLKO WINDOWS), a następnie odkomentować właściwą konfigurację.
   UWAGA – podać własne ścieżki dostępu do plików

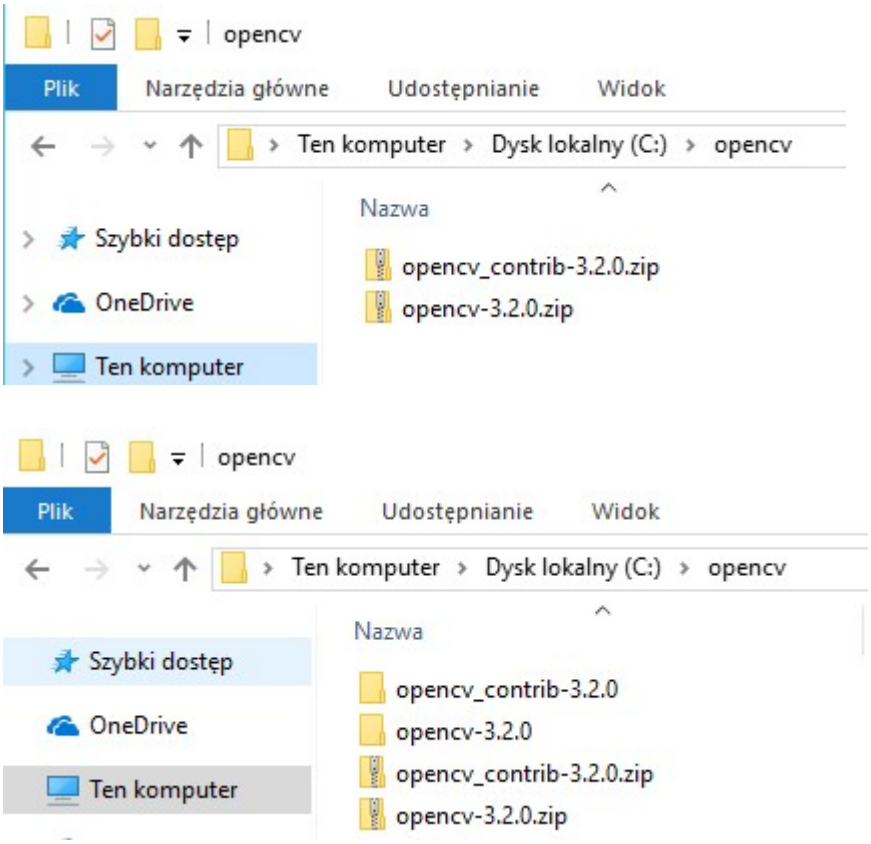
```
INCLUDEPATH += "D:\Lib\opencv\build\include"

#Dla oddzielnych bibliotek dla modułów
LIBS += -L"D:\Lib\opencv\build\x64\vc14\lib"
LIBS += -lopencv_core320d \
        -lopencv_highgui320d \
        -lopencv_imgcodecs320d\
        -lopencv_imgproc320d \
        -lopencv_video320d\
        -lopencv_videoio320d

#Dla zbudowanego opencv_world (ściągnięte binarki ze strony OpenCV)
#LIBS += -L"C:\Opencv_3_2\opencv\build\x64\vc14\lib"
#LIBS += -lopencv_world320d
```



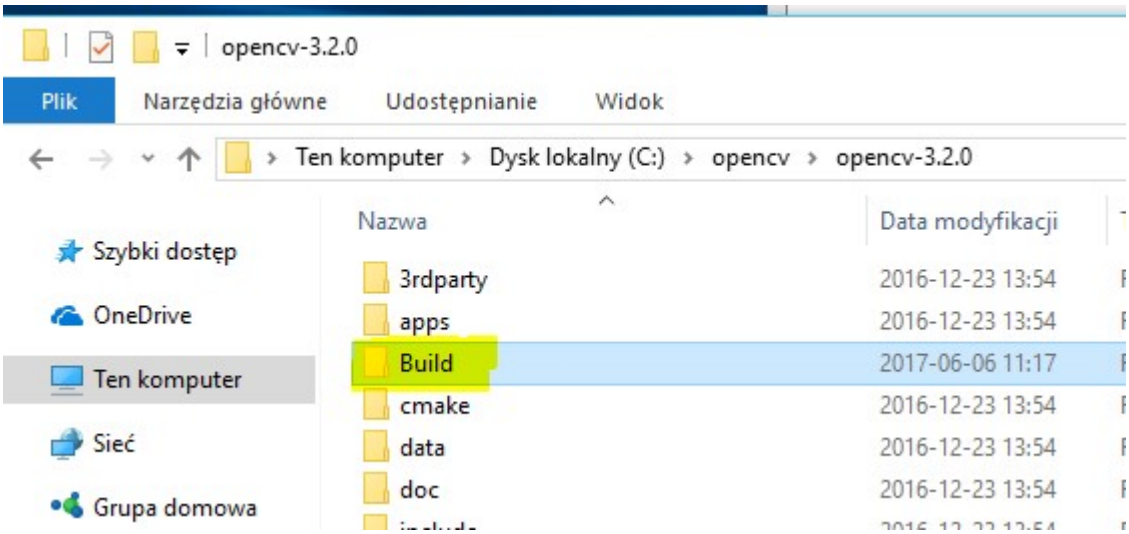
2. Budowanie ze źródeł z wykorzystaniem CMAKE
    a. Ze strony https://github.com/opencv/opencv/releases ściągnąć kod źródłowy głównego pakietu OpenCV (https://github.com/opencv/opencv/archive/3.2.0.zip)
    b. Ze strony https://github.com/opencv/opencv_contrib/releases/tag/3.2.0 ściągnąć kod źródłowy pakietów dodatkowych (https://github.com/opencv/opencv_contrib/archive/3.2.0.zip)

    !!!UWAGA!!! –kod źródłowy można też pobrać bezpośrednio z repozytoriów – dalszy proces instalacji będzie dokładnie ten sam

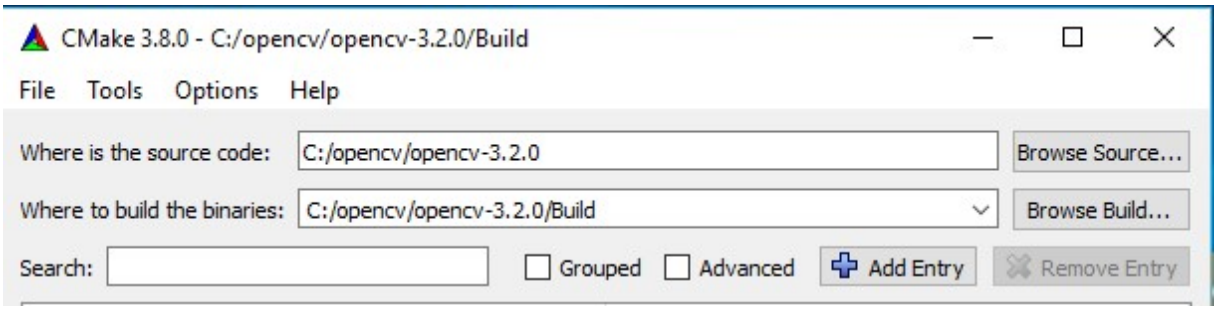
c. Umieścić pliki w odpowiednim katalogu i rozpakować



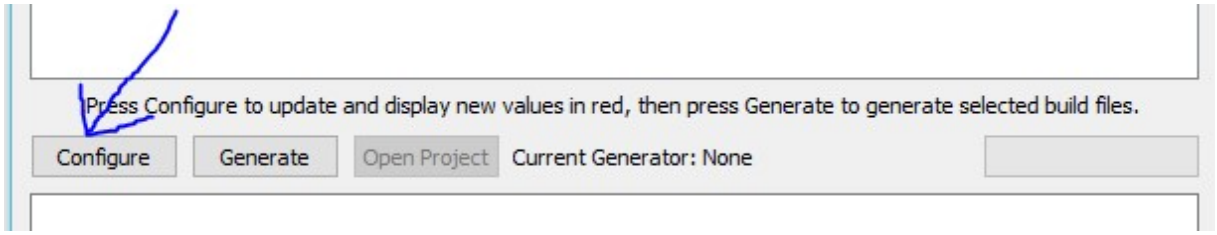
d. Stworzyć katalog Build

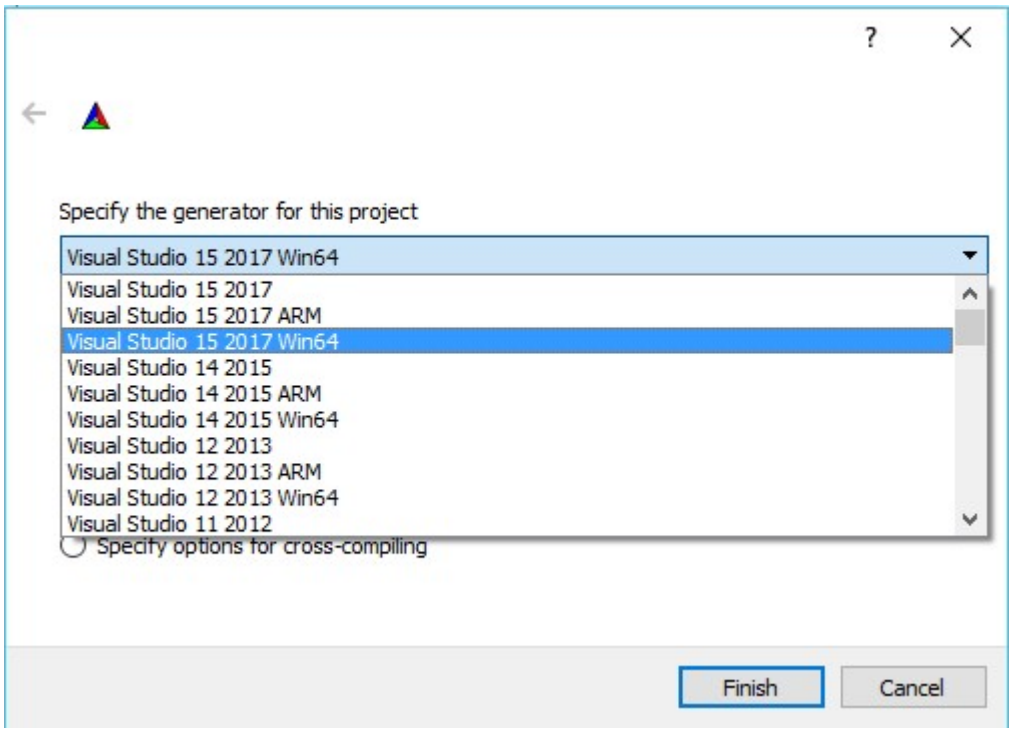


e. Uruchomić program CMake (cmake-gui) i podać ścieżki dostępu do źródeł

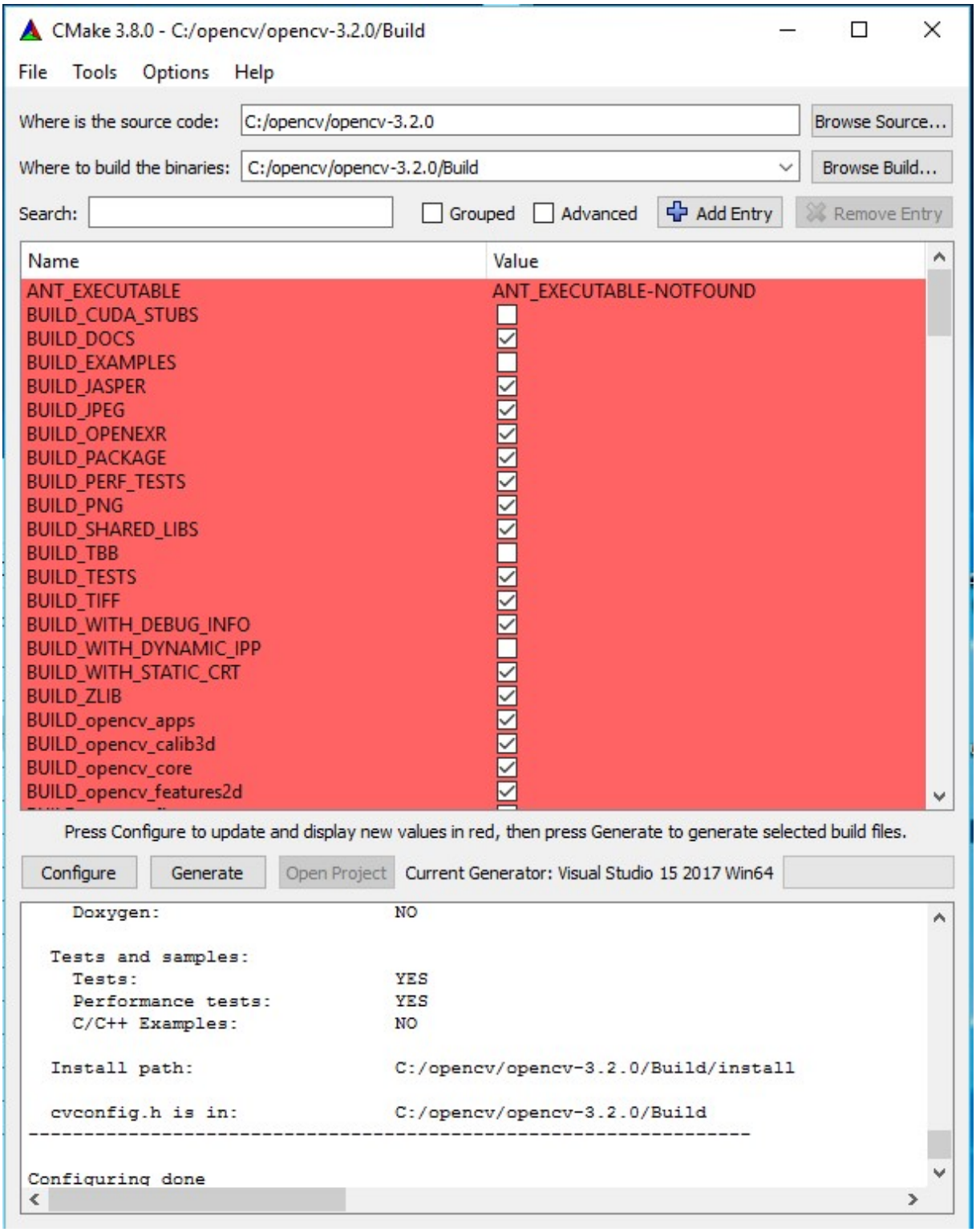


f. Wcisnąć przycisk configure i wybrać kompilator (przykład pokazuje wybór przy zainstalowanym Visual Studio 2017). Poczekać na zakończenie procesu konfiguracji

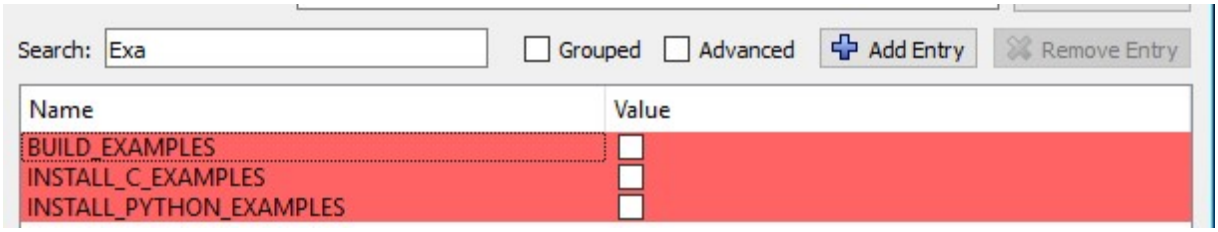


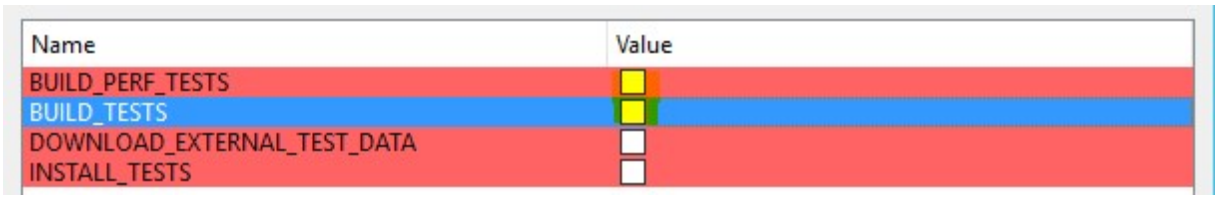


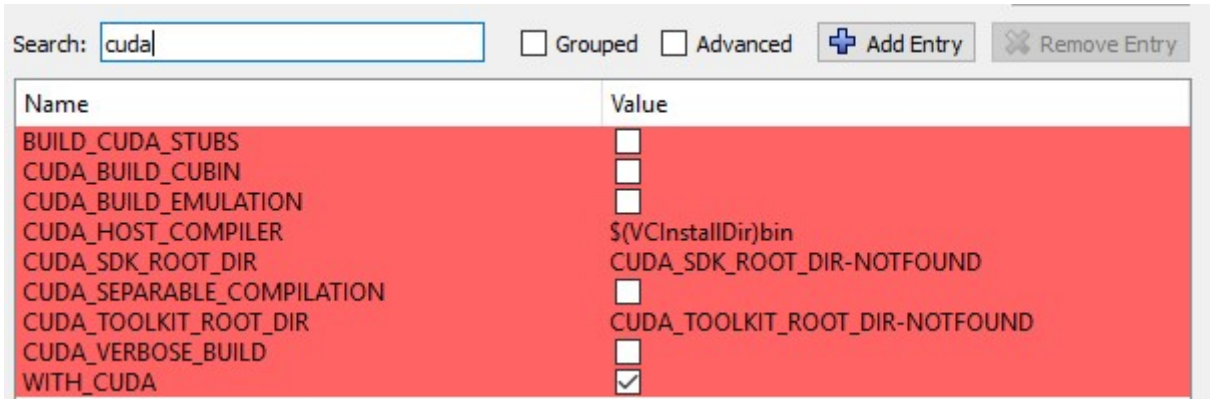
(**czerwone pola nie oznaczają błędów, tylko opcje nieskonfigurowane lub z wartościami domyślnymi**)



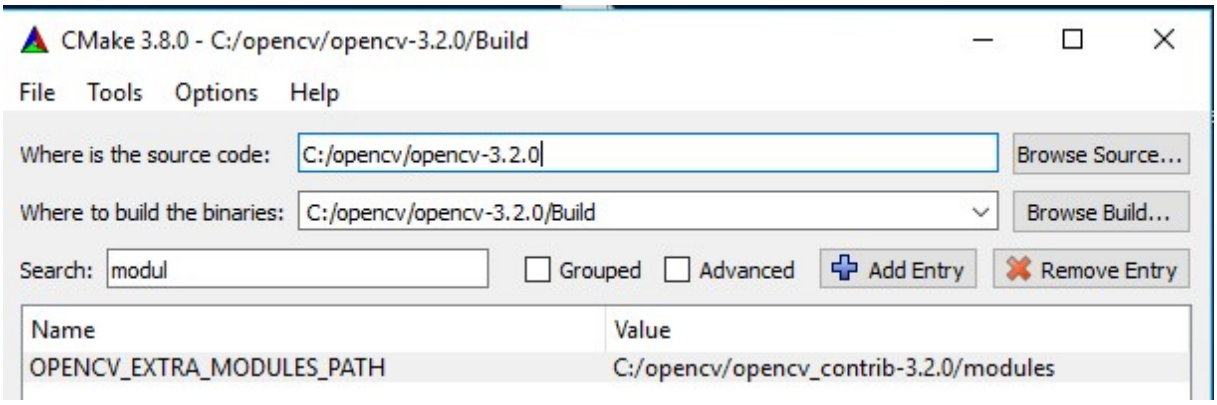
g. Wybrać opcje konfiguracji (na zrzutach ekranu pokazano przykładową konfigurację, przygotowaną do jak najszybszego budowania bibliotek – wyłączone zostało budowanie testów oraz przykładów).



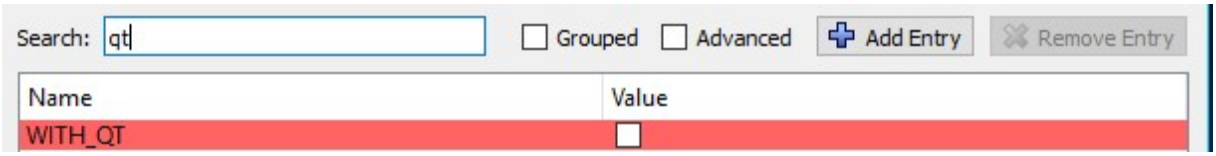




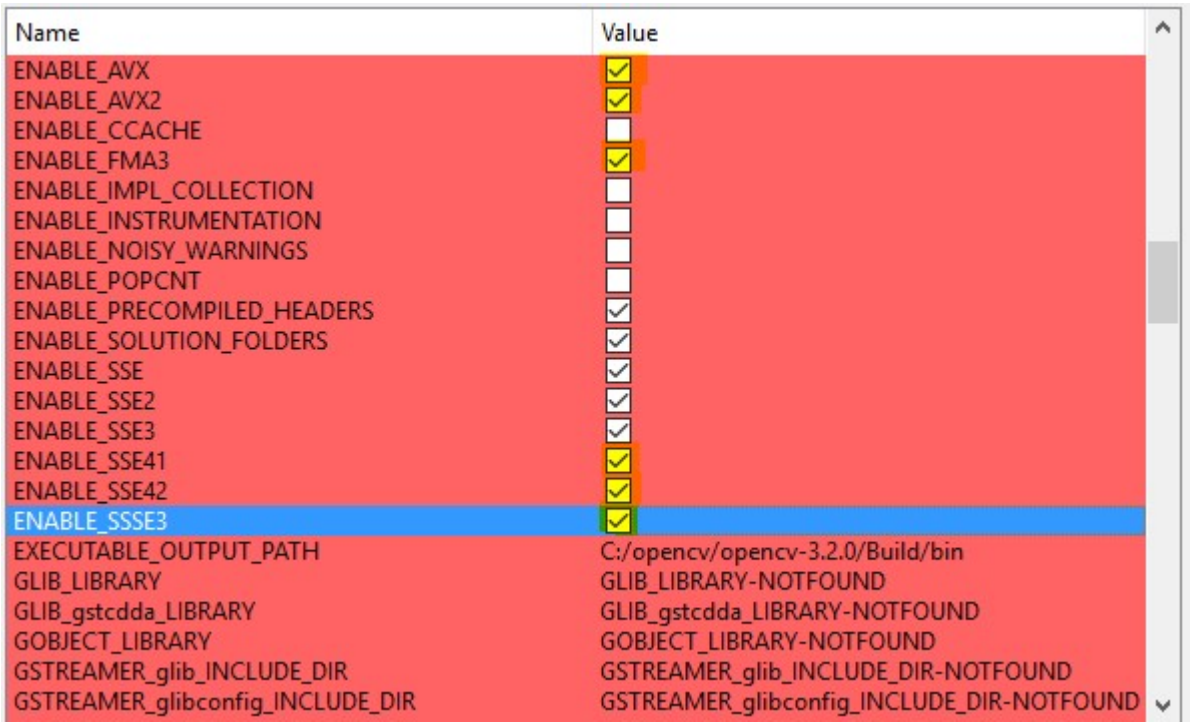
h. Podać ścieżkę do modułów dodatkowych (katalog modules w rozpakowanym opencv_contrib)



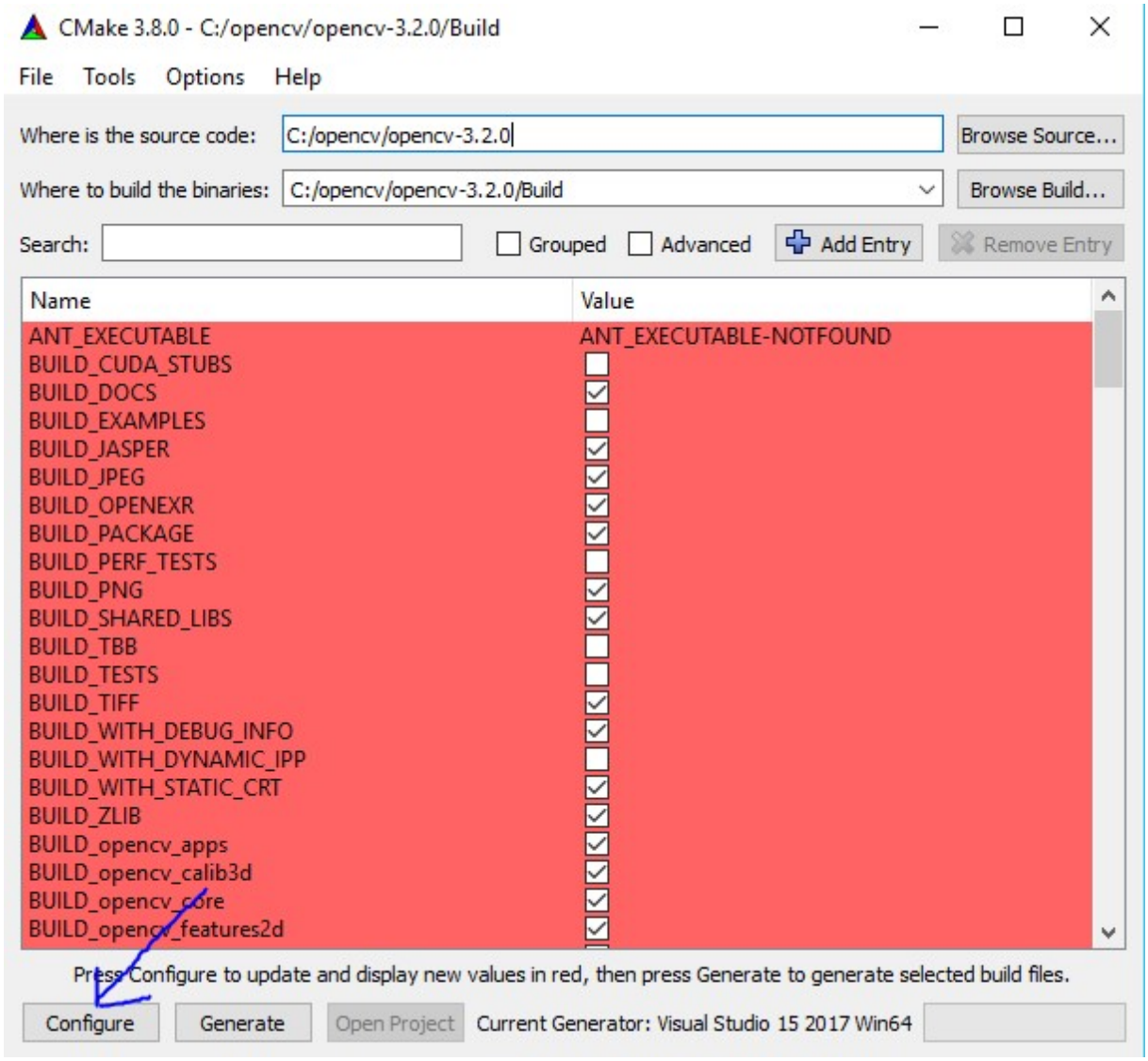
i. OpenCV można zbudować z tzw. QT backend lub bez – nie powinno to mieć wpływu na działanie

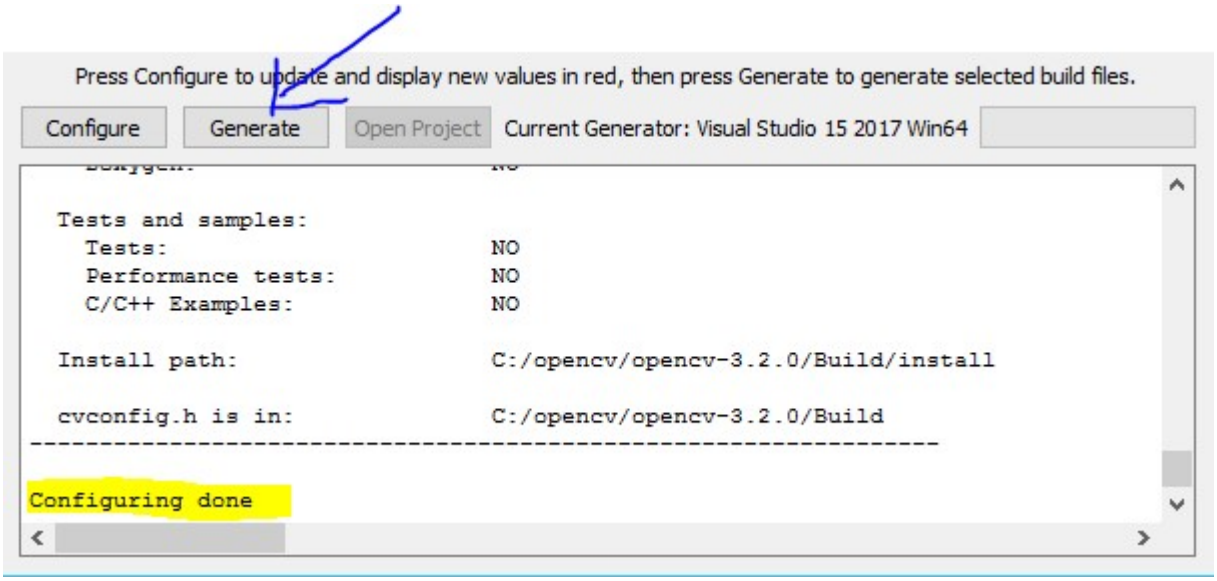


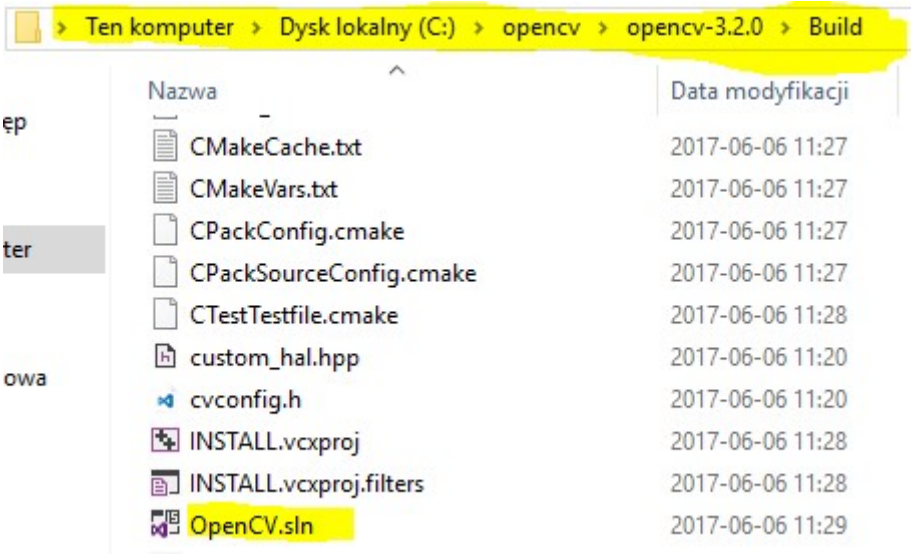
j. Opcje zwiększające wydajność – **tylko na nowszych komputerach, z procesorami Intel. W przypadku braku pewności pozostawić ustawienia domyślne**.



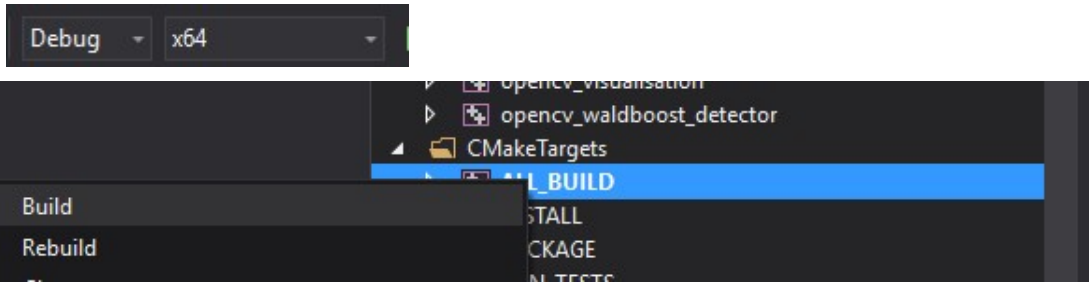
k. Po ustawieniu parametrów jeszcze raz nacisnąć przycisk Configure, a następnie Generate, co spowoduje wygenerowanie plików potrzebnych do kompilacji i budowania bibliotek

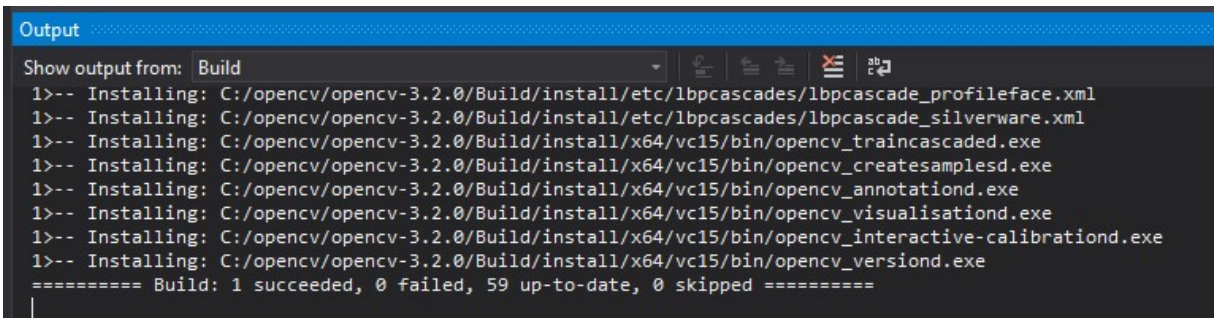


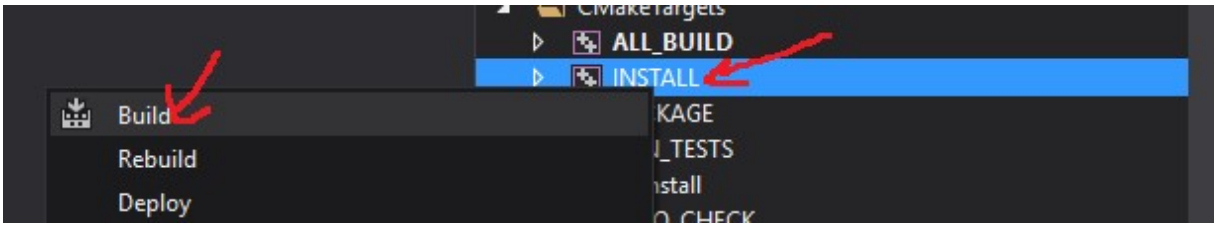


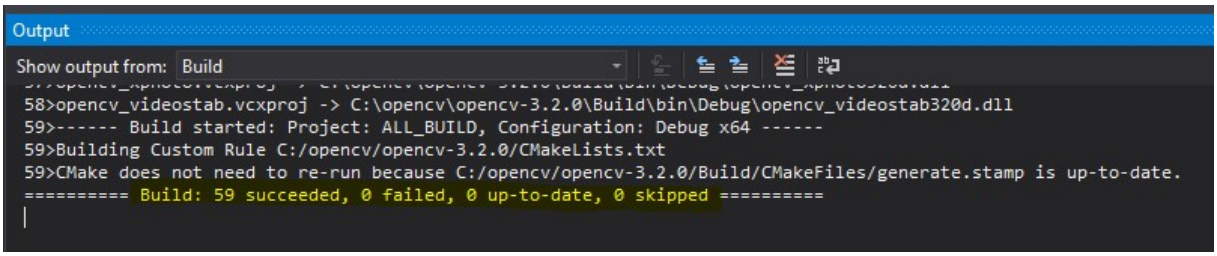


I. Wybrać właściwą konfigurację i zbudować projekty w odpowiedniej kolejności

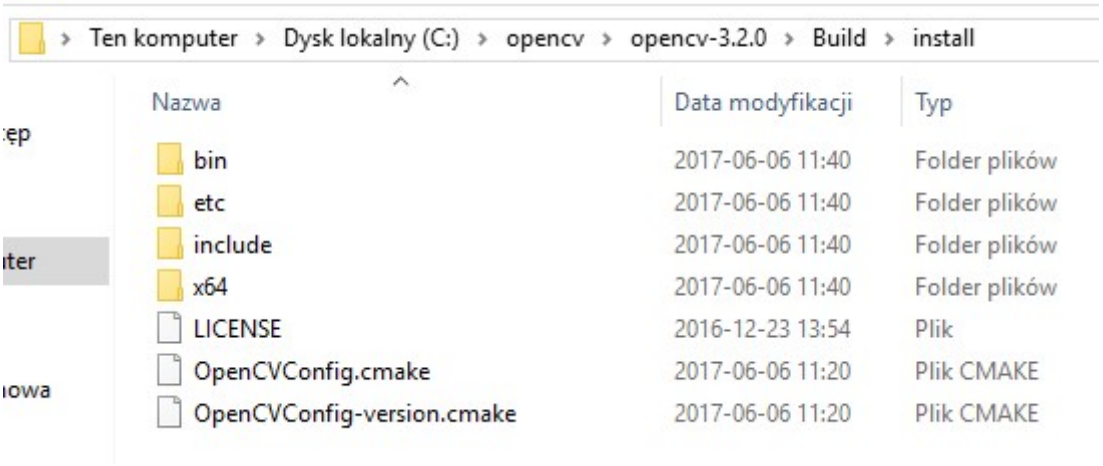


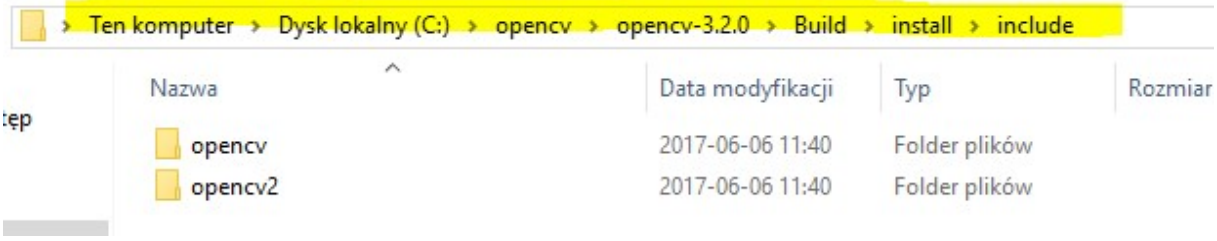


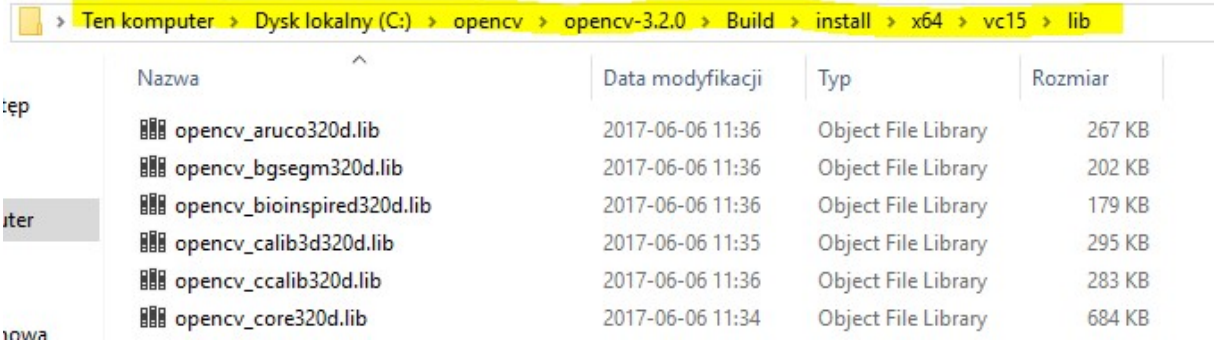




m. Do katalogu install zostaną skopiowane pliki wynikowe (biblioteki i pliki nagłówkowe)







Mac

OpenCV jest dostępne tylko przez zbudowanie bibliotek z kodów źródłowych z wykorzystaniem CMAKE

Z poziomu terminala:

```
sudo mkdir -p /opt/src
sudo chown $(whoami):staff /opt
sudo chown $(whoami):staff /opt
sudo chown $(whoami):staff /opt/src
cd /opt/src
curl -L https://github.com/opencv/opencv/archive/3.2.0.zip -o opencv32.zip
curl -L https://github.com/opencv/opencv_contrib/archive/3.2.0.zip -o opencv32contrib.zip
unzip opencv32.zip
unzip opencv32contrib.zip
mv -v opencv-3.2.0 /opt/src/opencv32
mv -v opencv_contrib-3.2.0 /opt/src/opencv32_contrib
cd /opt/src/opencv32
mkdir /opt/src/opencv32/release
cd /opt/src/opencv32/release
cmake \
-D CMAKE_INSTALL_PREFIX=/opt/opencv32 \
-D OPENCV_EXTRA_MODULES_PATH=/opt/src/opencv32_contrib/modules \
-D BUILD_TIFF=ON \
-D BUILD_opencv_java=OFF \
-D WITH_CUDA=OFF \
-D ENABLE_FAST_MATH=1 \
-D ENABLE_AVX=ON \
-D WITH_OPENGL=ON \
-D WITH_OPENCL=ON \
-D WITH_IPP=OFF \
-D WITH_TBB=ON \
-D WITH_EIGEN=ON \
-D WITH_VTK=OFF \
-D WITH_V4L=OFF \
-D BUILD_TESTS=OFF \
-D BUILD_PERF_TESTS=OFF \
-D CMAKE_BUILD_TYPE=RELEASE \    (lub DEBUG;RELEASE)

make -j8
make install
```

Graficznie:

COMING SOON

Linux:

Tak jak w przypadku MacOS budowanie tylko ze źródeł, z wykorzystaniem CMAKE

http://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/

http://docs.opencv.org/trunk/d7/d9f/tutorial_linux_install.html

https://www.linuxhint.com/how-to-install-opencv-on-ubuntu/