

# INSTALACJA OPENCV v 3.2

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

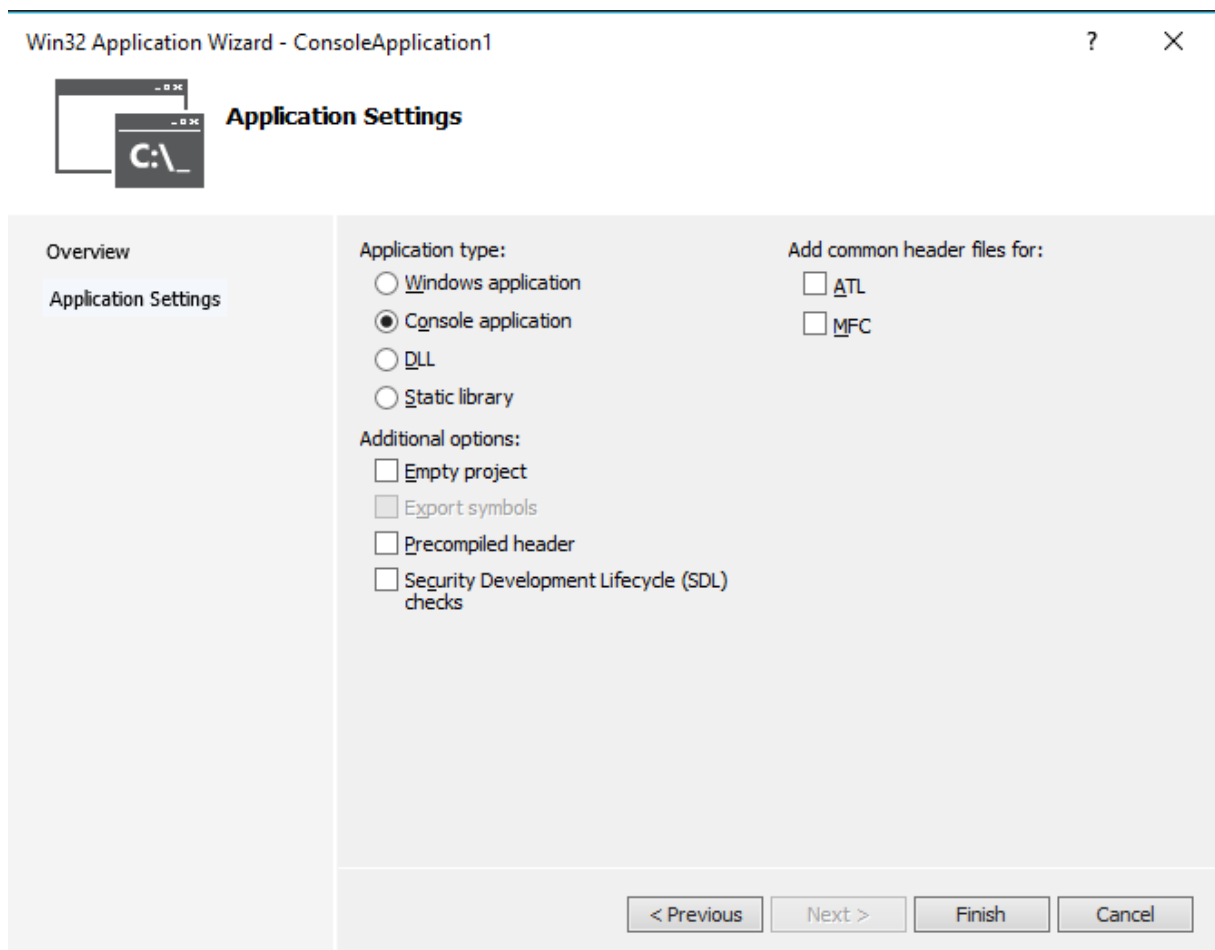
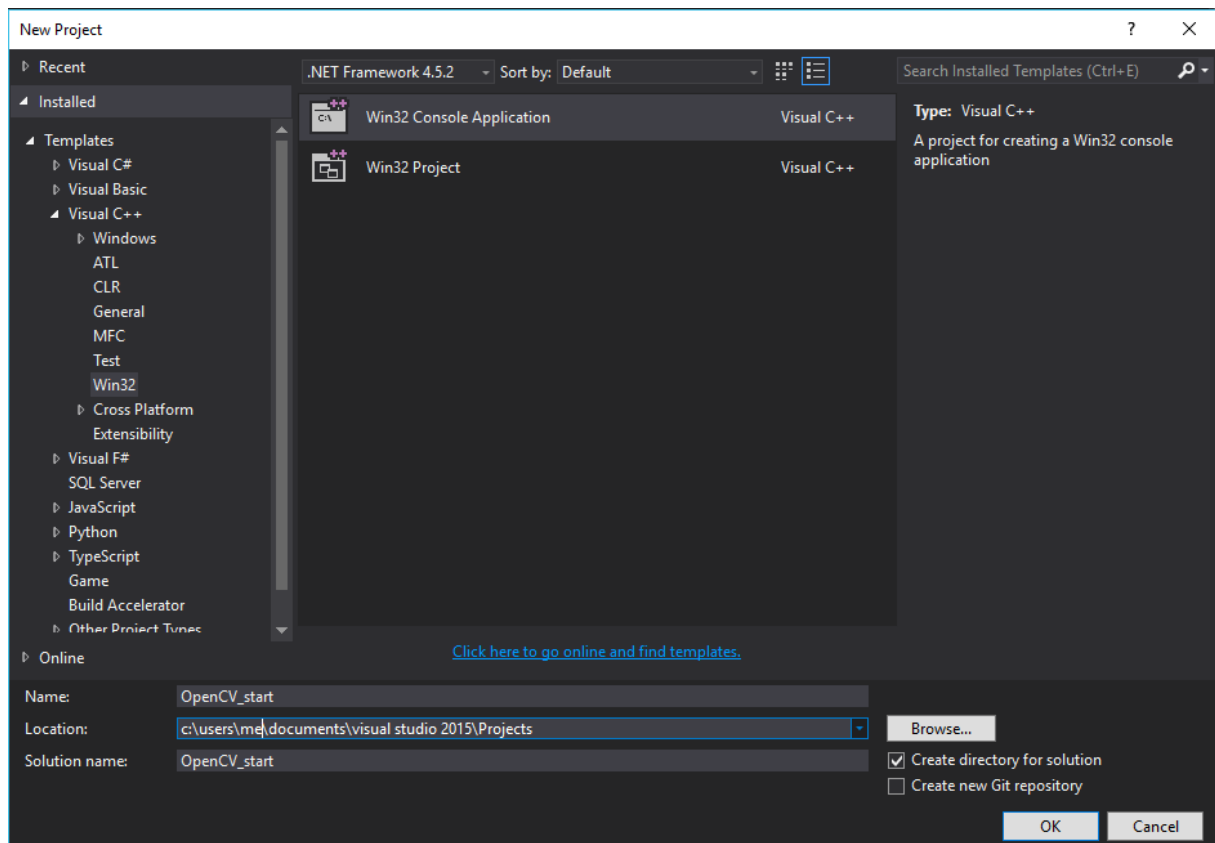
## Windows

### 1. Skorzystanie z wersji release (binarnej)

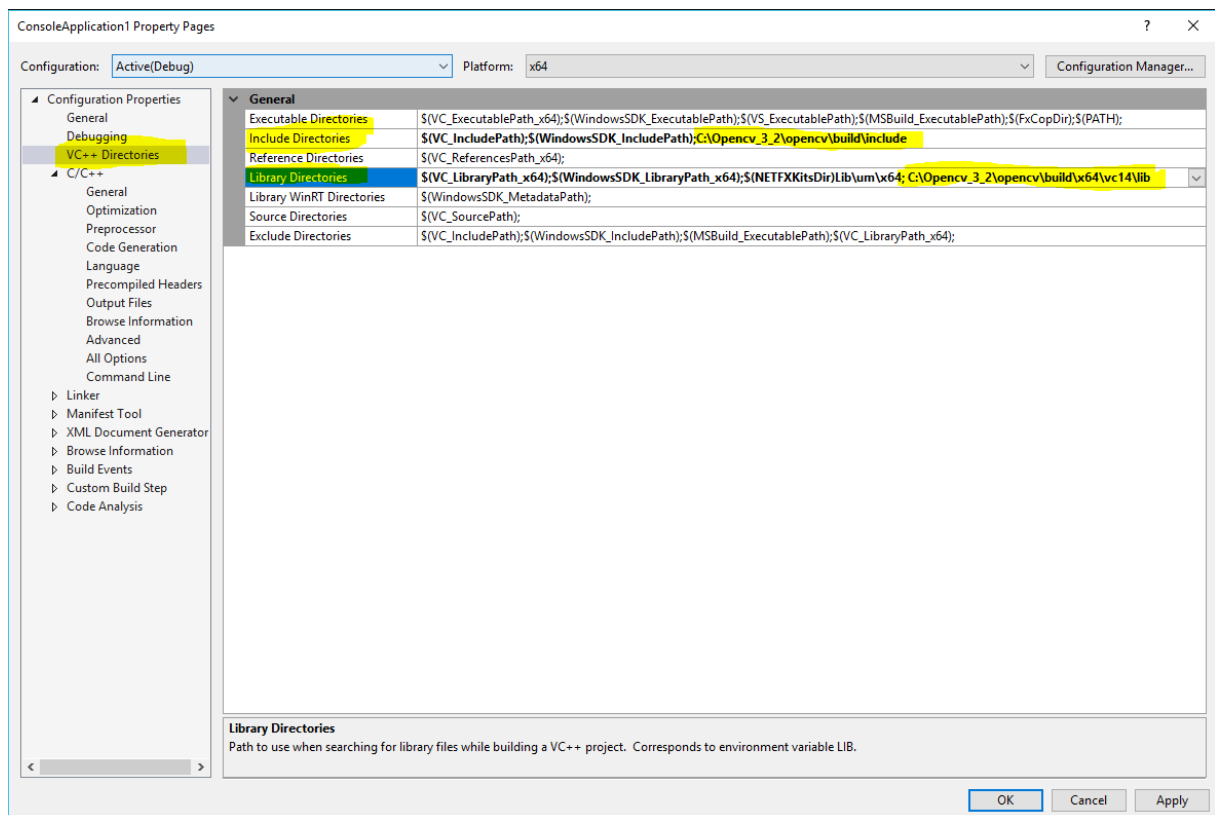
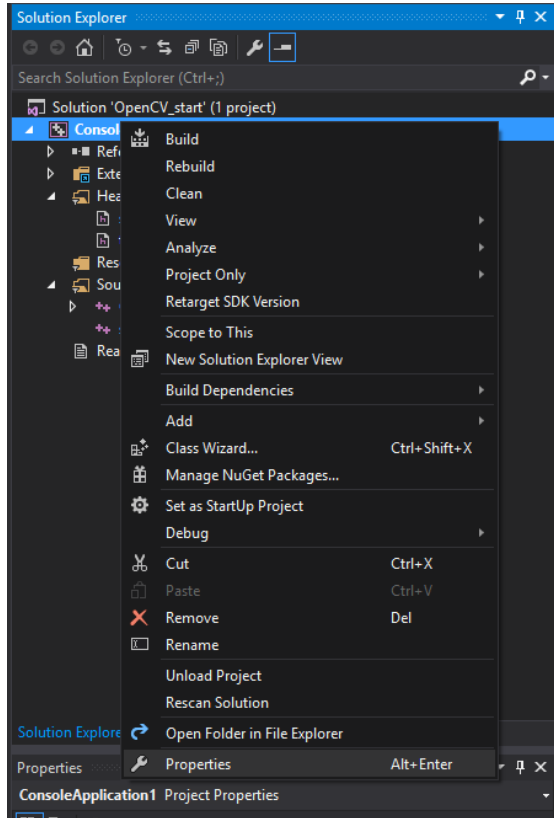
- Ze strony <http://opencv.org/releases.html> ściągnąć wersję Win pack (<https://sourceforge.net/projects/opencvlibrary/files/opencv-win/3.2.0/opencv-3.2.0-vc14.exe/download>) – biblioteki kompatybilne z Visual Studio od wersji 2015 oraz minGW, w wersji 64 bit
- Rozpakować plik do dowolnego katalogu (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

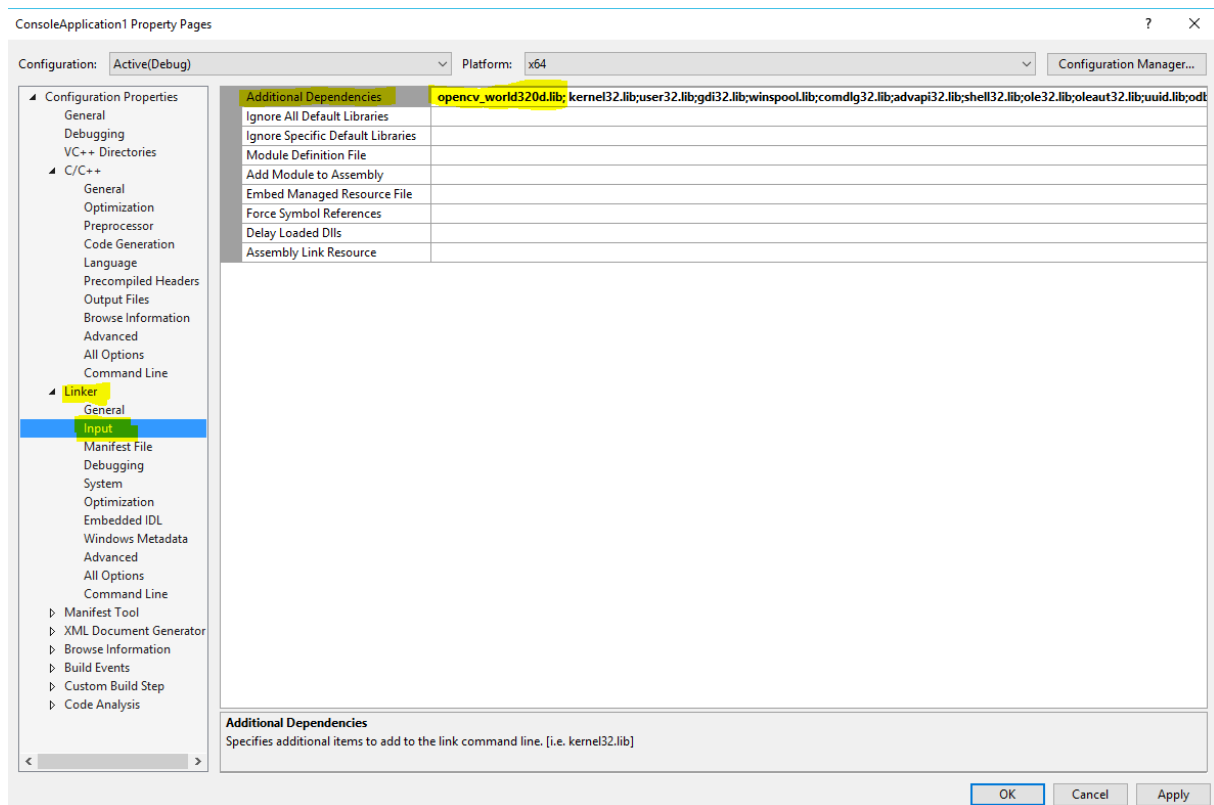
Konfiguracja na przykładzie Visual Studio:

### 1. Utworzenie nowego projektu



## 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
```

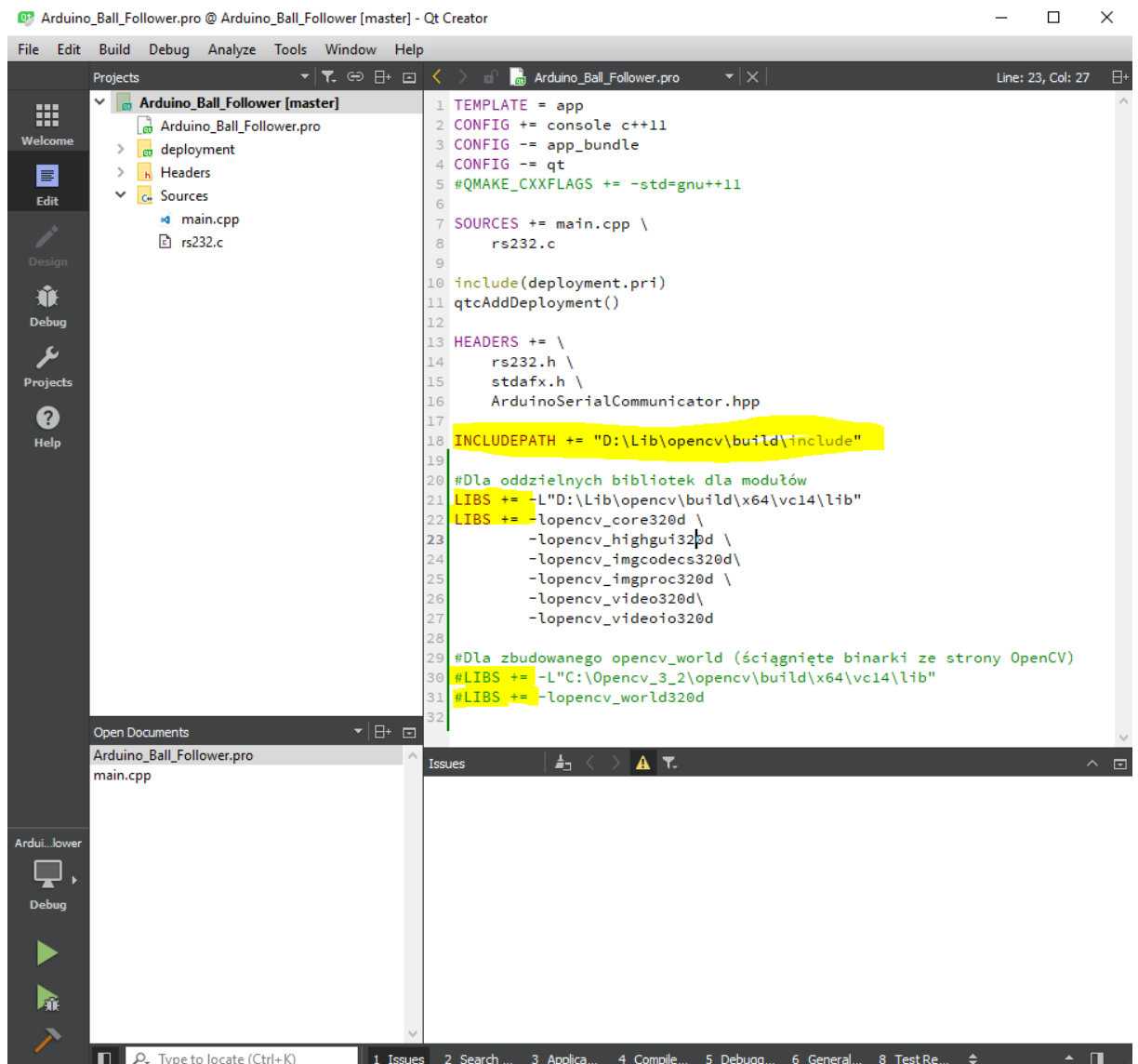
```
LIBS += -L"D:\Lib\opencv\build\x64\vc14\lib"
```

```
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

COMING SOON

## 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](http://docs.opencv.org/trunk/d7/d9f/tutorial_linux_install.html)

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

