

Install OpenCV 4.5.5 for C++, Windows 10, Code::Blocks, TDM-GCC-64



Sourabh Jigjinni · [Follow](#)

5 min read · Nov 29, 2018



202



26



• • •

This is a step-by-step installation of OpenCV 4.5.5 on Windows. I was inspired to make this installation guide with the help of screenshots, to make it as easy as possible for new users to get started. Thanks to [Zahid Hasan for the guide on version 3.2.0](#) which most of this is based on.

• • •

1. Install Code::Blocks

Code::Blocks

The IDE with all the features you need, having a consistent look, feel and operation across platforms.

- News
- Features
- Downloads
- User manual
- Forums
- Wiki
- License
- Donations

GPL ✓ W3C ✓ GetFirefox ✓ SOURCEFORGE ✓ Support this project

sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03-setup.exe

Code::Blocks / Downloads / Binary releases

Binary releases

Please select a setup package depending on your platform:

- Windows XP / Vista / 7 / 8x / 10
- Linux 32 and 64-bit
- Mac OS X

NOTE: For older OSes use older releases. There are releases for many OS version and platforms on the [Sourceforge.net](#) page.

NOTE: There are also more recent nightly builds available in the [forums](#) or (for Ubuntu users) in the [Ubuntu PPA repository](#). Please note that we consider nightly builds to be stable, usually.

NOTE: We have a [Changelog for 20.03](#) that gives you an overview over the enhancements and fixes we have put in the new release.

NOTE: The default builds are 64 bit (starting with release 20.03). We also provide 32bit builds for convenience.



File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

NOTE: The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

NOTE: The codeblocks-20.03mingw-setup.exe file includes additionally the GCC/G++/GFortran compiler and GDB debugger from [MinGW-W64 project](#) (version 8.1.0, 32/64 bit, SEH).

Download [Code::Blocks](#) here, or google codeblocks > downloads > Download the binary Release.

Click on the [sourceforge.net](#) link for the *codeblocks-20.03-setup.exe* option as we want to use TDM (64 bit) compiler.

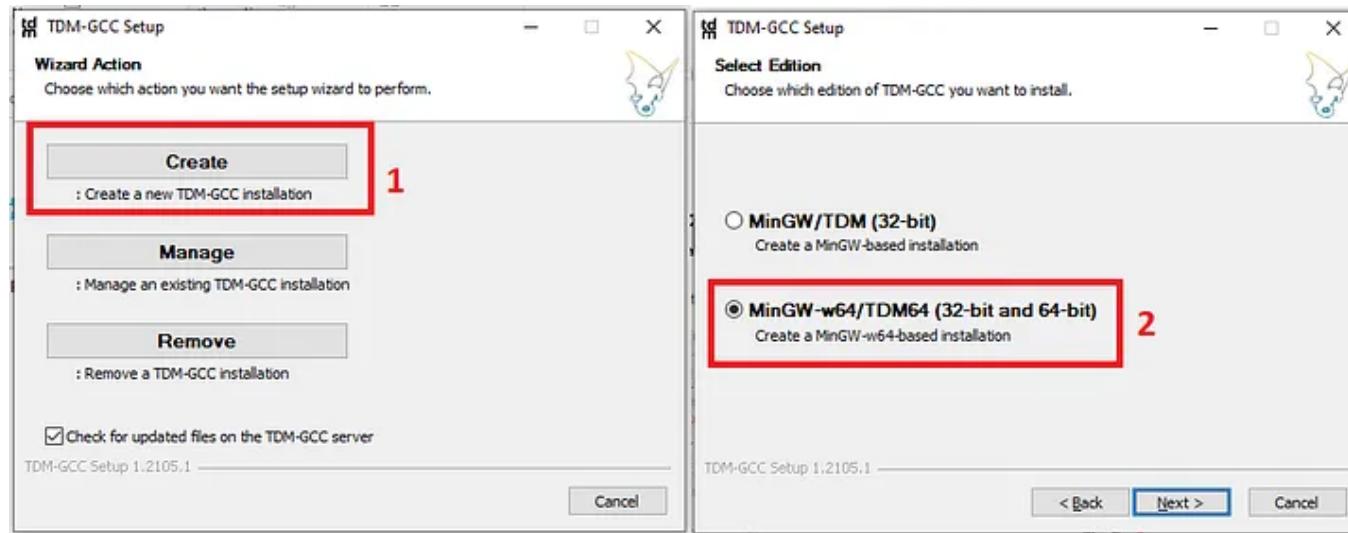
• • •

2. Install TDM-GCC

Download [TDM-GCC](#) here. Make sure to select the 64 bit version.

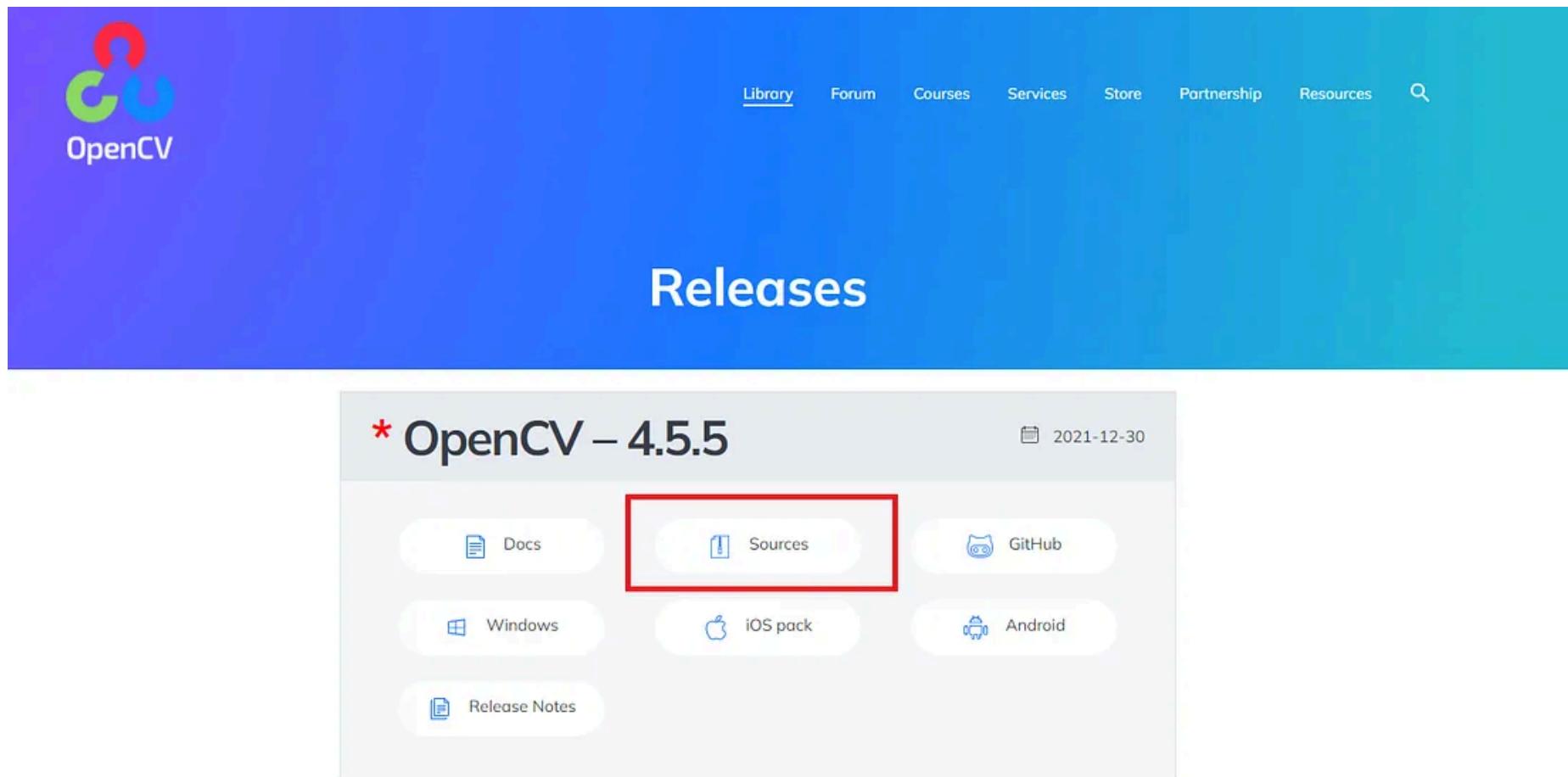


Install it in C:\ drive. it will look like C:\TDM-GCC-64\



3. Download the source of OpenCV 4.5.5

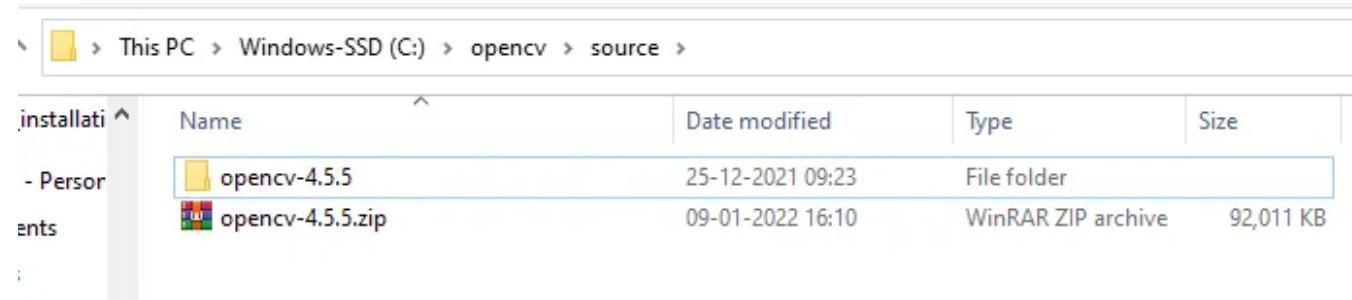
Download [OpenCV 4.5.5](#) and click sources to get the zip file.



Create folders:

- C:\opencv\source\
- C:\opencv\build\

Copy/move the *opencv-4.5.5.zip* file inside the **C:\opencv\source** folder, and unzip it here.



Optional Step: If you need extra-modules, we can download them by following steps mentioned [here](#). I usually use the Aruco module for tag detection. Ignore this if you are new.

4. Install CMake

Download [CMake](#) and install.



About ▾ Resources ▾ Developer Resources ▾ Download PURCHASE SUPPORT

Latest Release (3.22.1)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the linux-x86_64 tar file is all under the directory cmake-linux-x86_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.rst at the top of the source tree. See also the [CMake 3.21 Release Notes](#).

Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.22.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.22.1.zip

Binary distributions:

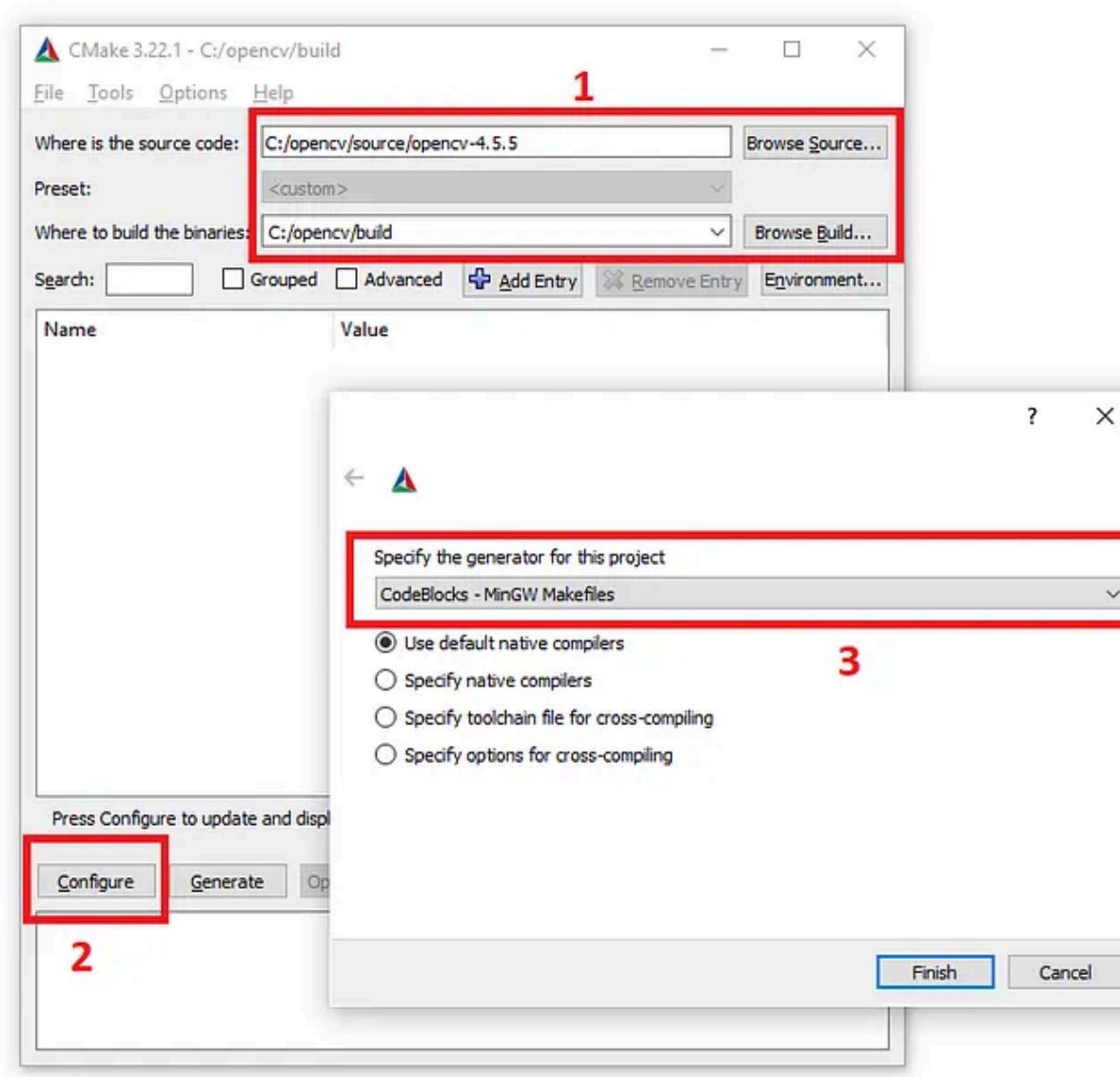
Platform	Files
Windows x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.22.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.22.1-windows-x86_64.zip
Windows i386 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.22.1-windows-i386.msi
Windows i386 ZIP	cmake-3.22.1-windows-i386.zip
macOS 10.13 or later	cmake-3.22.1-macos-universal.dmg
	cmake-3.22.1-macos-universal.tar.gz

5. Build the Binaries

1. Open cmake, set source path to C:\opencv\source\ and binary path to C:\opencv\build.

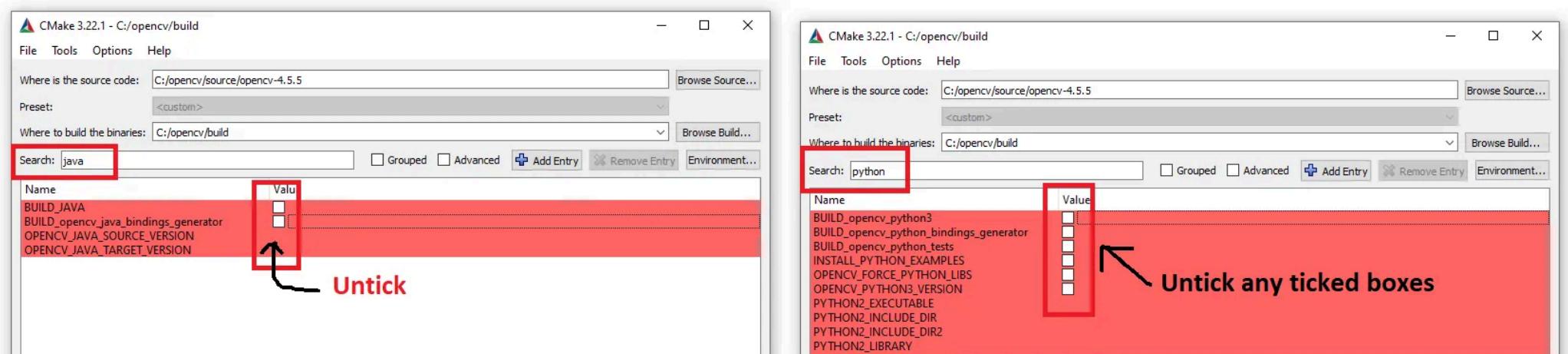
2. Click configure

3. Choose CodeBlock – MinGW Makefiles (Should be set by default)

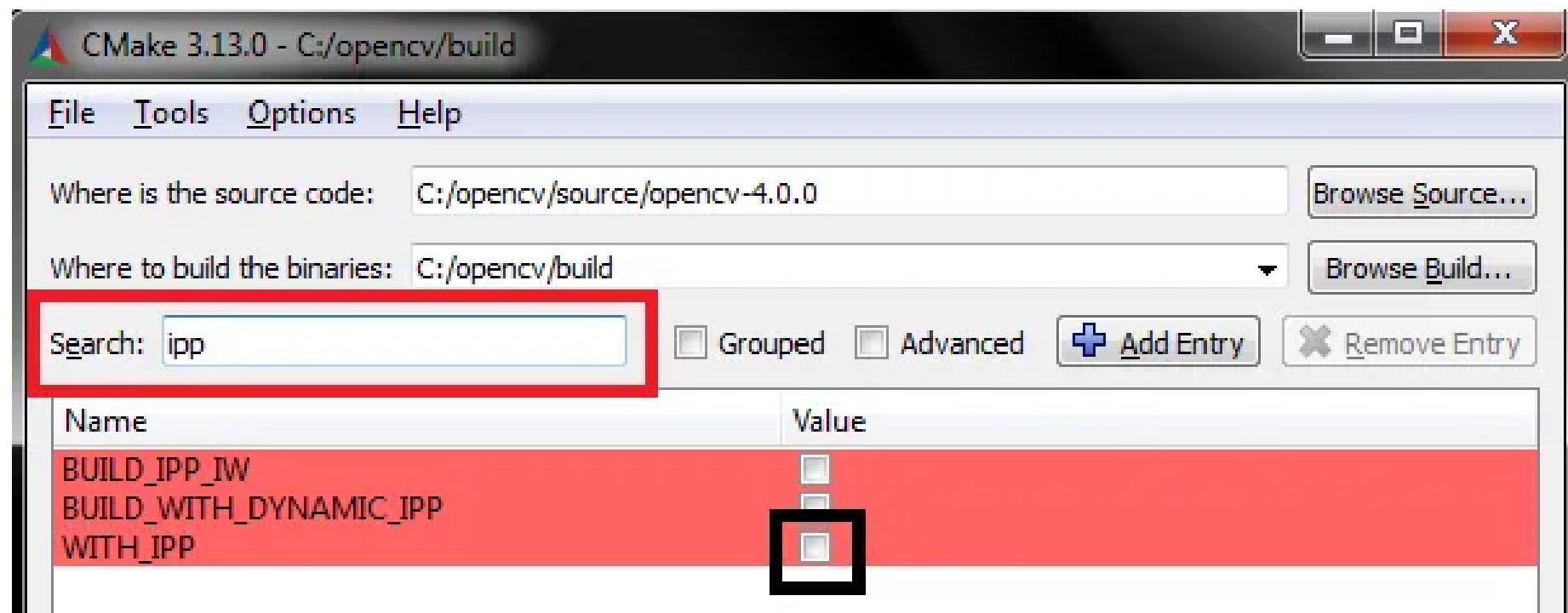


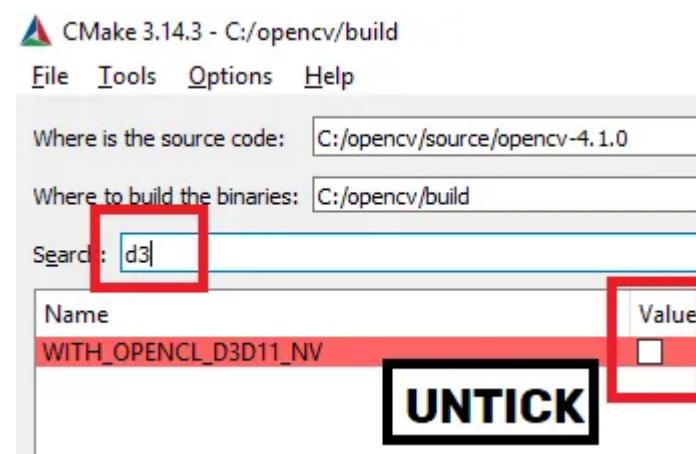
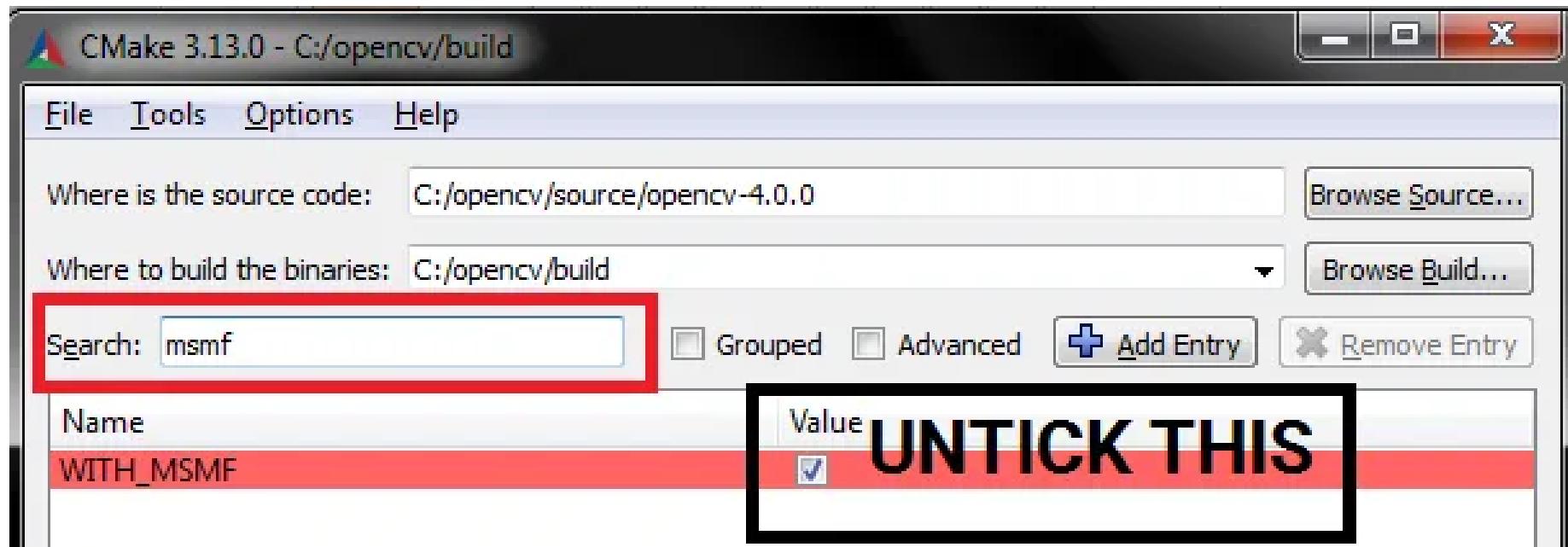
After configuring you will see options in red. We need to disable some of these to build the system:

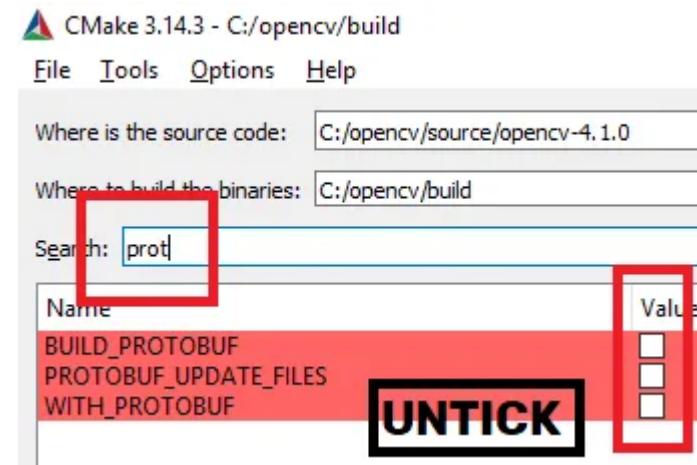
Firstly I usually disable all python and Java bindings. Type into the search bar:



- untick WITH_MSMF (media foundation needs special win sdk, only available for VS)
- ENABLE_PRECOMPILED_HEADERS=OFF (untick)
- WITH_IPP=OFF WITH_TBB=OFF (again libs available are for VS only) (untick)







MAKE SURE

- WITH_MFMS=OFF (UNTICK)
- WITH_IPP=OFF (UNTICK)
- WITH_TBB=OFF (UNTICK)
- ENABLE_PRECOMPILED_HEADERS=OFF (UNTICK)

And

- WITH_OPENCL=ON
- WITH_OPENCL_D3D11_NV=OFF (UNTICK)

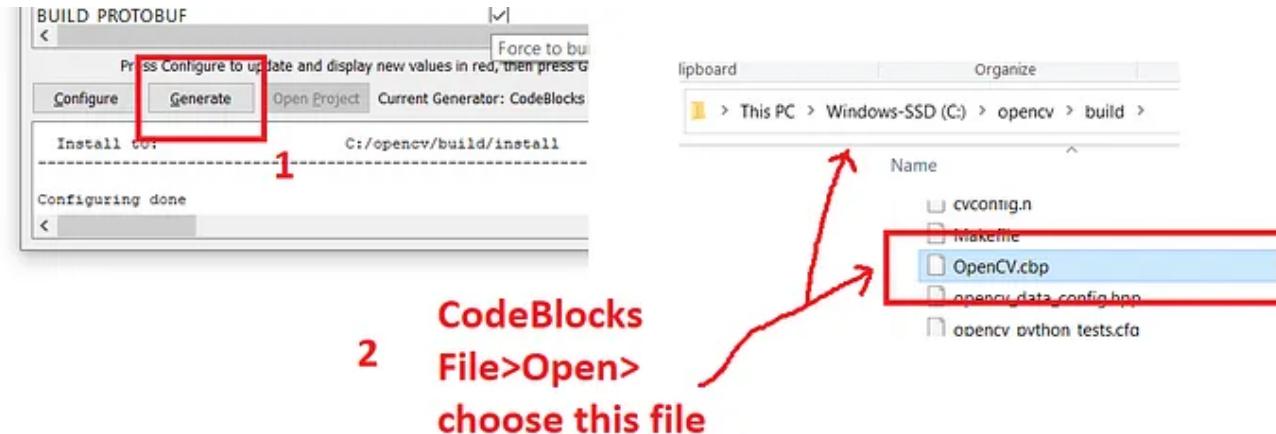
- WITH_DIRECTX=ON

Lastly

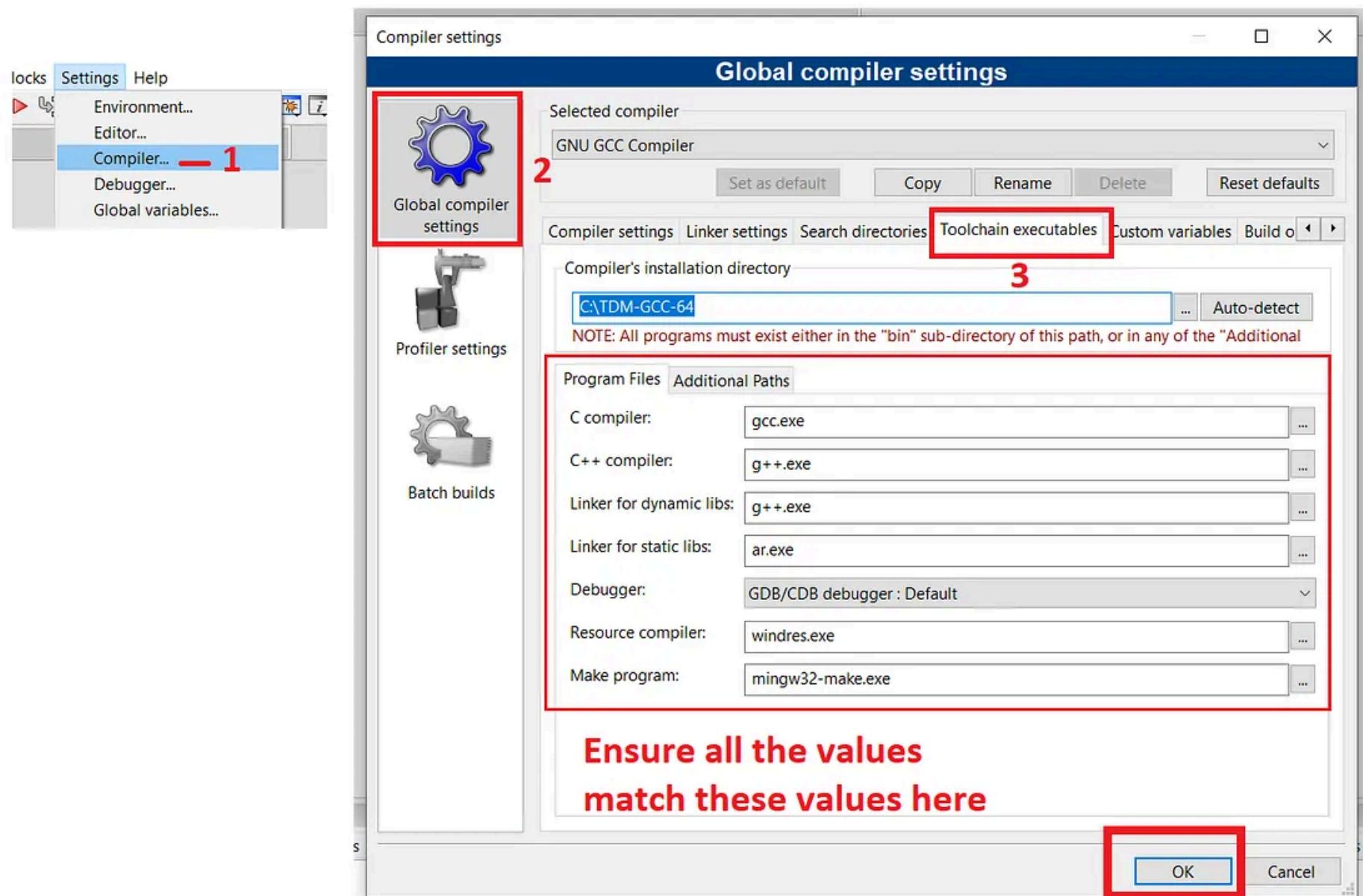
- BUILD_PROTOBUF = OFF (UNTICK)
- PROTOBUF_UPDATE_FILES = OFF (UNTICK)
- WITH_PROTOBUF = OFF (UNTICK)
- OPENCV_ENABLE_ALLOCATOR_STATS = OFF (UNTICK)

Optional Step follow up: If you downloaded the extra-modules, now is the time to enable them. Refer [here](#) for ALL modules. Refer [here](#) for only Aruco module.

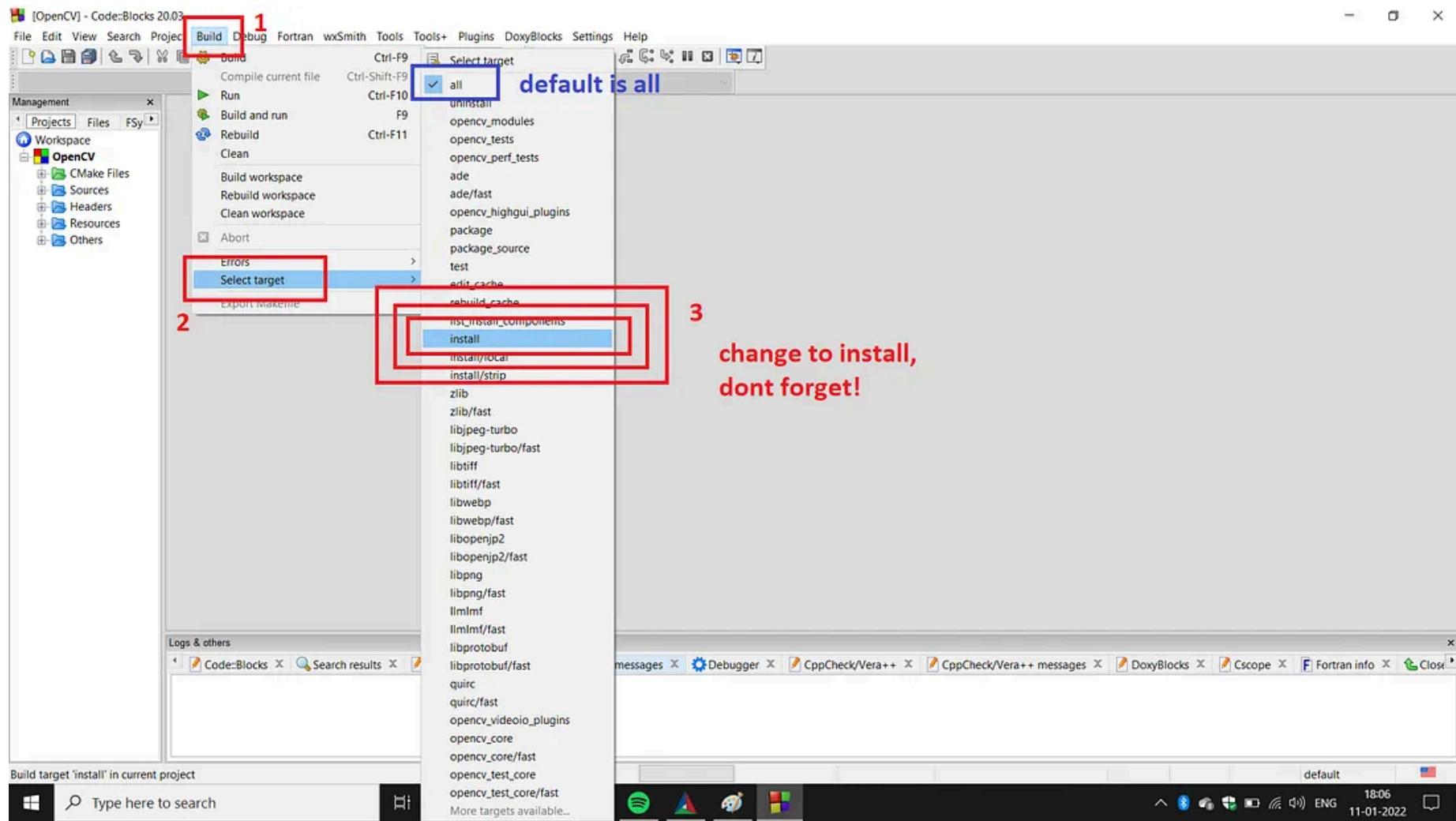
Next click Generate.



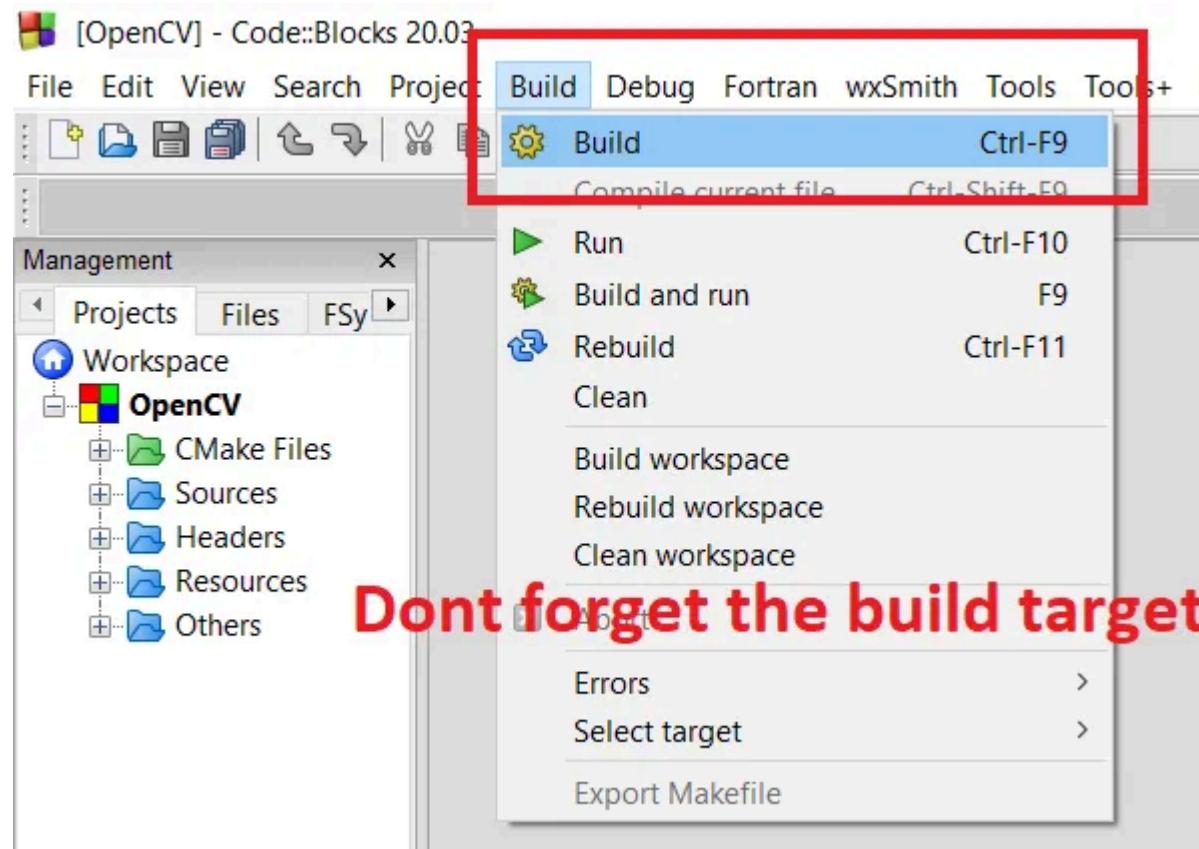
1. You will find a codeblocks project file (opencv.cbp) in C:\opencv\build folder. Just double click it and codeblocks should load it. If it doesn't just find the codeblocks app and open it.
2. Go to ‘settings’, choose ‘compiler’ and click ‘Toolchain executable’. In the ‘compiler’s installation directory’ field choose the “bin” folder of MinGW C:\TDM-GCC-64\bin. set the following:
 - c compile: gcc.exe
 - c++ compiler: g++.exe
 - Linker for dynamic libs: ar.exe



3. DONT BUILD TARGET IN A HURRY, this is where I went wrong, while following Zahid Hasan's article. Select build > select target > install

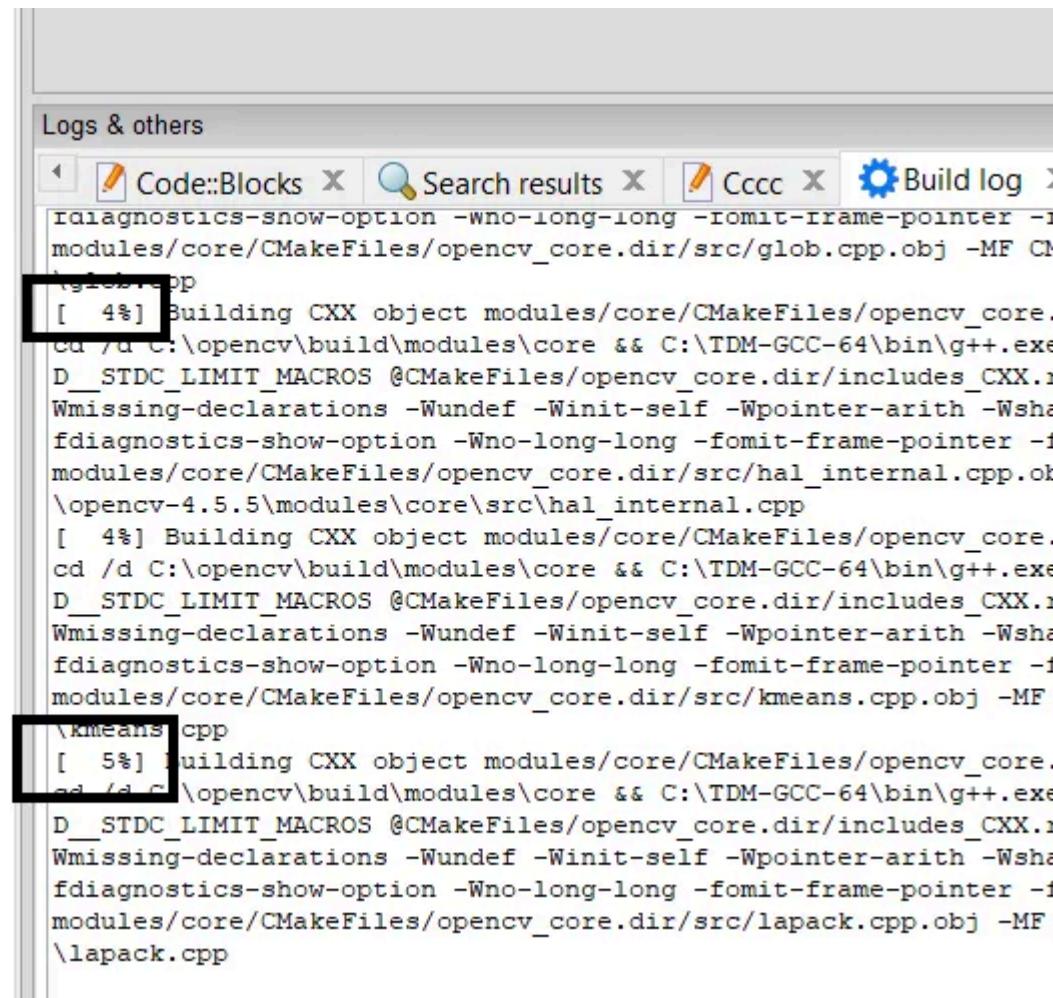


4. After this step you can build.



Tip: the percentage of the build done is shown here. This will take a while depending on your hardware.

If logs aren't visible by default, go to View > Logs, or press F2



```
raiagnostics-show-option -Wno-long-long -fomit-frame-pointer -fmodules/core/CMakeFiles/opencv_core.dir/src/glob.cpp.obj -MF CMglob.cpp
[ 4%] Building CXX object modules/core/CMakeFiles/opencv_core.dir/src/glob.cpp.o
cd /d C:\opencv\build\modules\core && C:\TDM-GCC-64\bin\g++.exe
D__STDC_LIMIT_MACROS @CMakeFiles/opencv_core.dir/includes_CXX.r
Wmissing-declarations -Wundef -Winit-self -Wpointer-arith -Wshadow
fdiagnostics-show-option -Wno-long-long -fomit-frame-pointer -fmodules/core/CMakeFiles/opencv_core.dir/src/hal_internal.cpp.o
\opencv-4.5.5\modules\core\src\hal_internal.cpp
[ 4%] Building CXX object modules/core/CMakeFiles/opencv_core.dir/src/kmeans.cpp.o
cd /d C:\opencv\build\modules\core && C:\TDM-GCC-64\bin\g++.exe
D__STDC_LIMIT_MACROS @CMakeFiles/opencv_core.dir/includes_CXX.r
Wmissing-declarations -Wundef -Winit-self -Wpointer-arith -Wshadow
fdiagnostics-show-option -Wno-long-long -fomit-frame-pointer -fmodules/core/CMakeFiles/opencv_core.dir/src/lapack.cpp.obj -MF
\kmeans.cpp
[ 5%] Building CXX object modules/core/CMakeFiles/opencv_core.dir/src/lapack.cpp.o
cd /d C:\opencv\build\modules\core && C:\TDM-GCC-64\bin\g++.exe
D__STDC_LIMIT_MACROS @CMakeFiles/opencv_core.dir/includes_CXX.r
Wmissing-declarations -Wundef -Winit-self -Wpointer-arith -Wshadow
fdiagnostics-show-option -Wno-long-long -fomit-frame-pointer -fmodules/core/CMakeFiles/opencv_core.dir/src/lapack.cpp.o -MF
\lapack.cpp
```

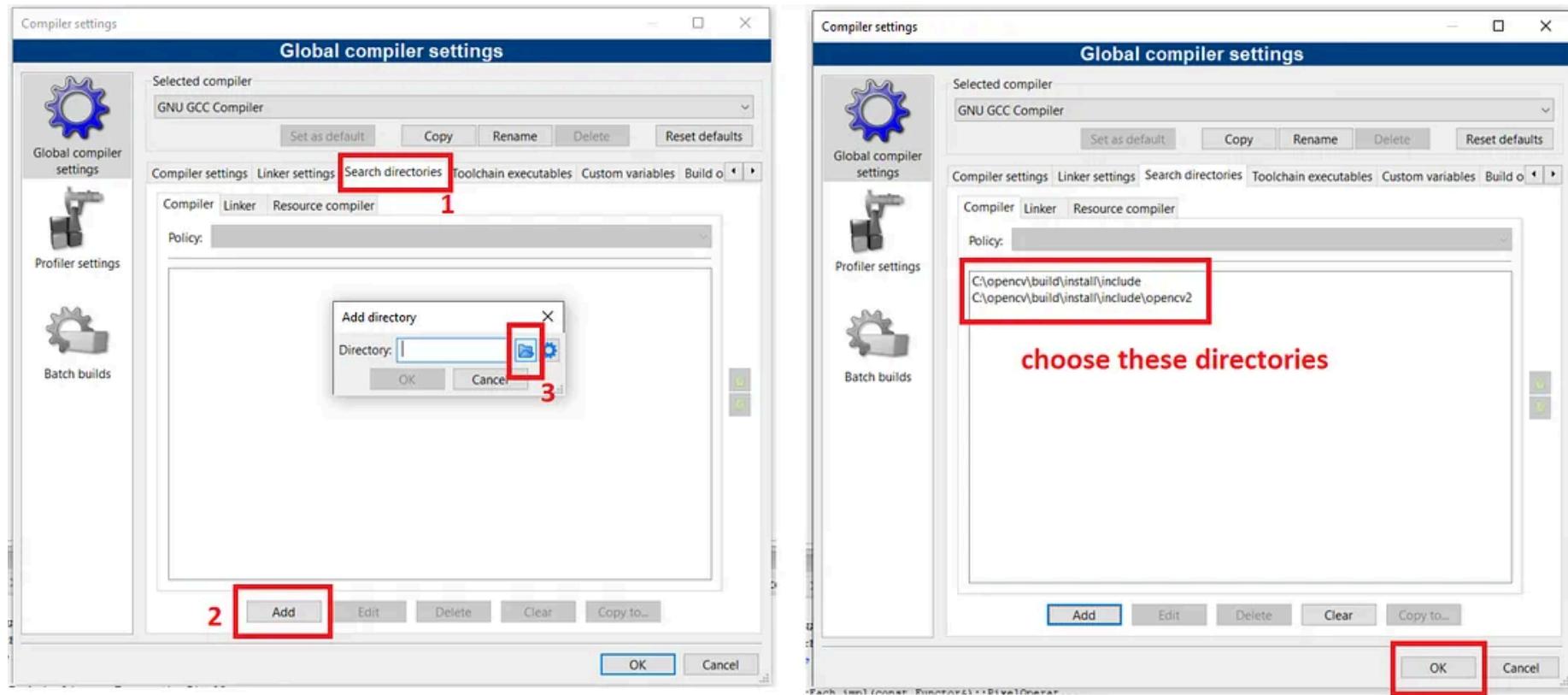
5. Add `C:\opencv\build\install\x64\mingw\bin` to the path.

- Add path variable in windows 10 with screenshots — [link](#)

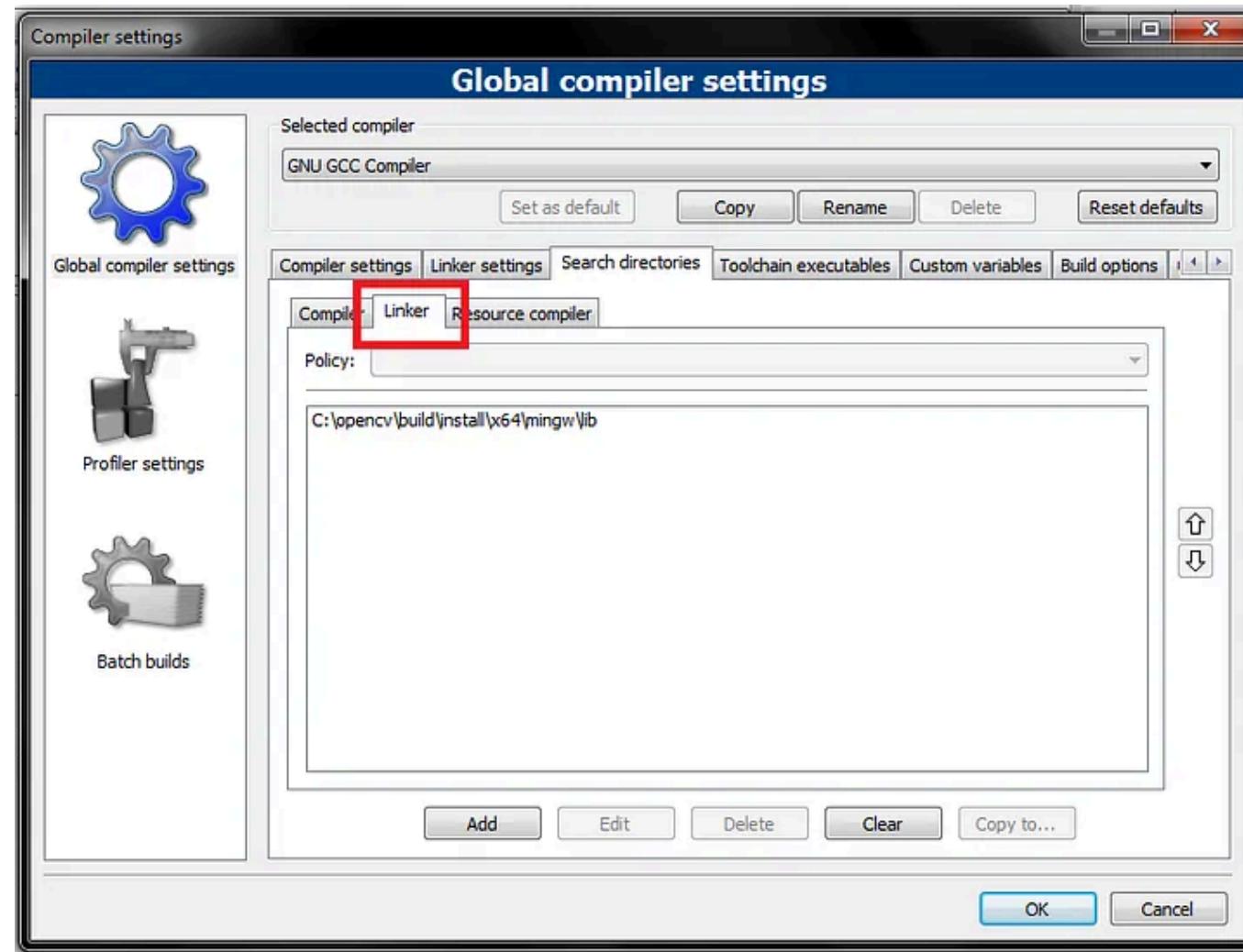
Tip: you can check path variables with `echo %path%;=&echo.%` in command prompt(cmd) or `$Env:Path.Split(';')` in powershell.

6. Run a C++ test program

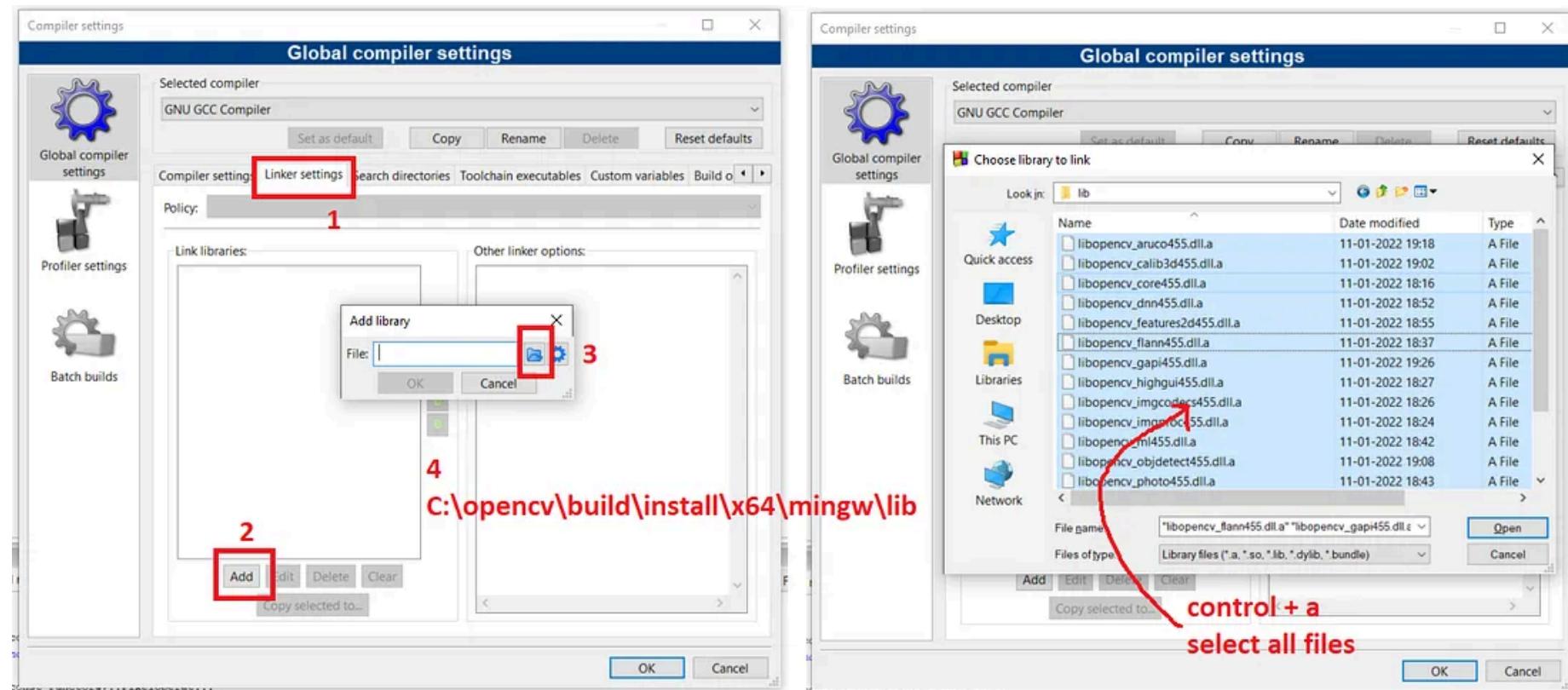
1. Create a test project in Code::Blocks. Select console application.
2. Go to settings -> compiler. Select ‘search directories’ and in the ‘compiler’ tab chose the followings:
 - C:\opencv\build\install\include
 - C:\opencv\build\install\include\opencv2



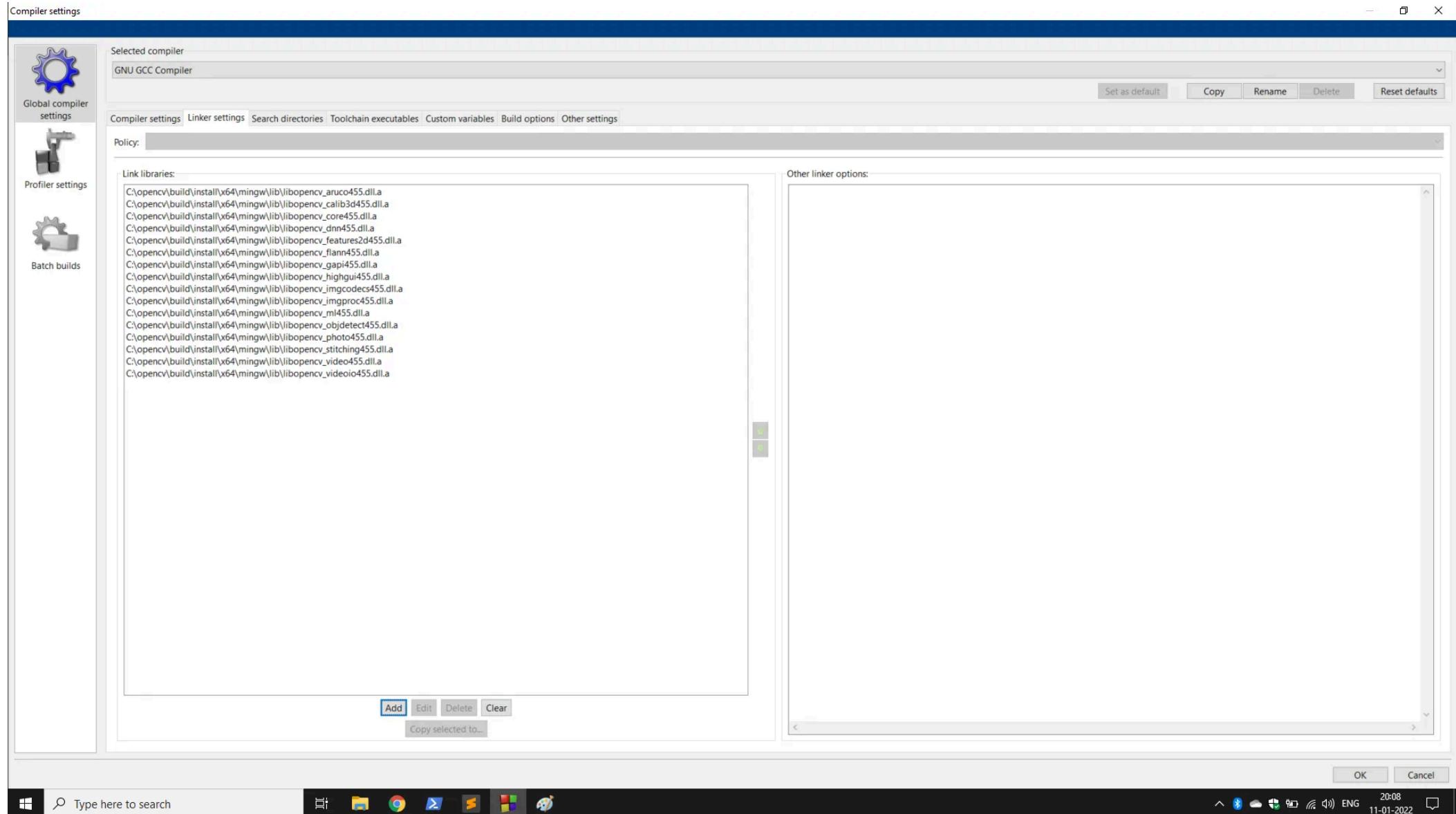
3. Select ‘Linker’ tab and add C:\opencv\build\install\x64\mingw\lib



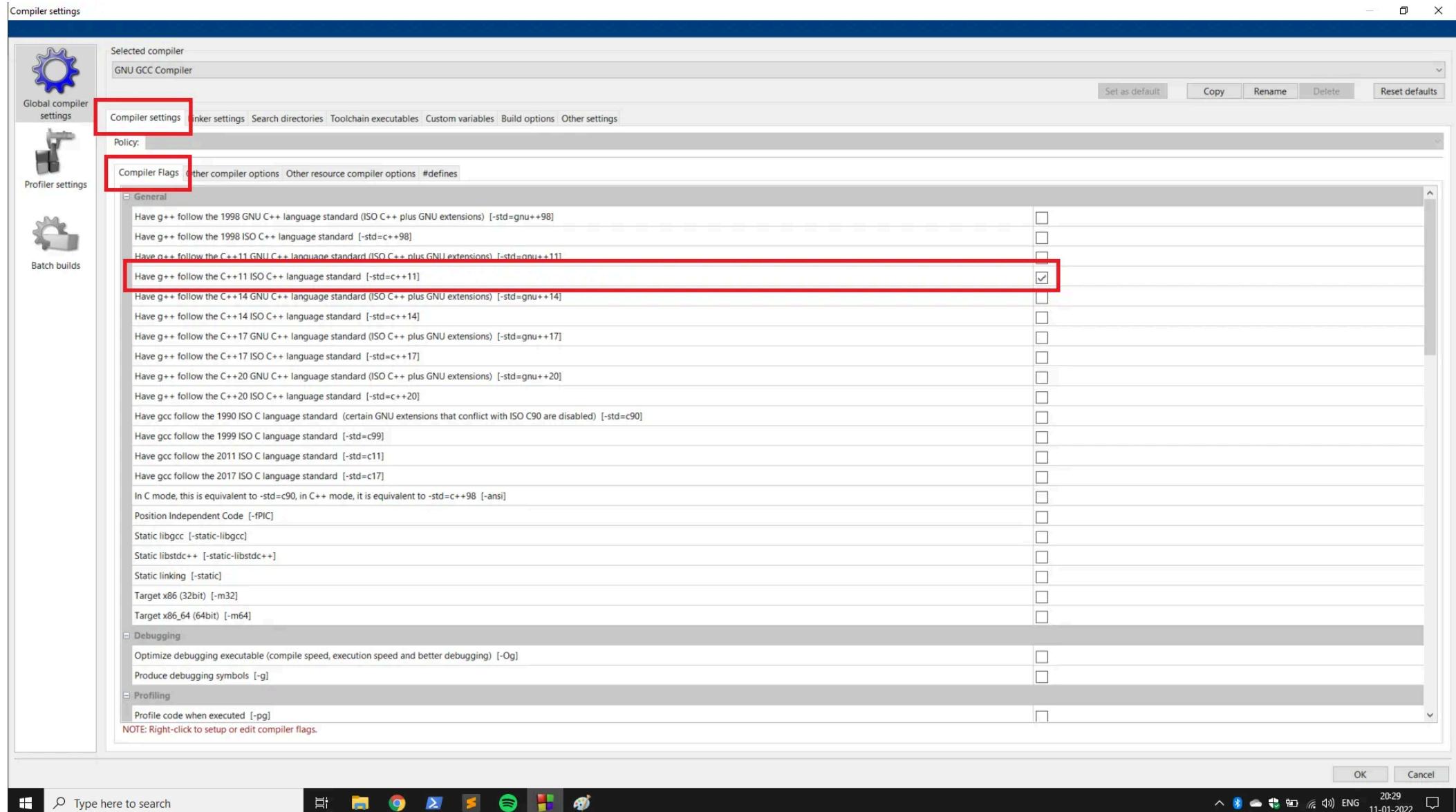
4. Go to 'Linker Settings' and add all the libraries from
C:\OpenCV\my_build\install\x64\mingw\lib folder



If it looks like this, all is good. Just one last step of using c++11 standard.



5. Set the compiler to c++11 standard (Settings -> Compiler)



6. Edit your main.cpp and add this:

```
1 #include <iostream>
2 #include <opencv2/highgui/highgui.hpp>
3 #include <opencv2/opencv.hpp>
4
5 using namespace std;
6 using namespace cv;
7
8 int main() {
9
10     VideoCapture cap(0);
11     if (!cap.isOpened()) {
12         cout << "Error initializing video camera!" << endl;
13         return -1;
14     }
15
16     char* windowName = "Webcam Feed";
17     namedWindow(windowName, WINDOW_AUTOSIZE);
18
19     while (1) {
20
```



Search



Write

Sign up

Sign in



```
21         imshow(windowName, frame);
22         if (waitKey(10) == 27) {
23             break;
24         }
25     }
26     cout << "Error reading frame from camera feed" << endl;
27     break;
28 }
29
30     case 27:
31         return 0;
32     }
33 }
```

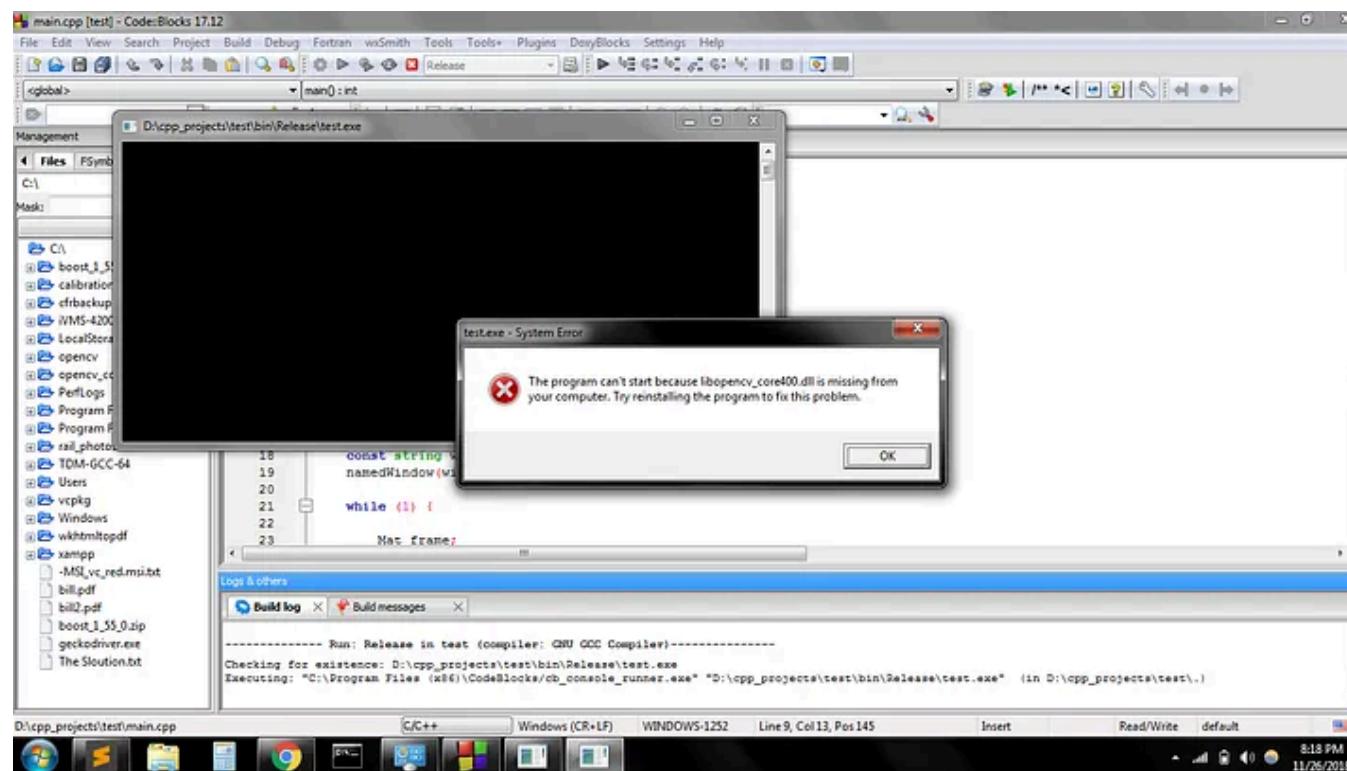
```
34     }
35     return 0;
36 }
```

start_webcam.cpp hosted with ❤ by GitHub

[view raw](#)

7. Build and run

If u get an error like this restart your computer.



If your program compiles your webcam should start. If u don't have a webcam try to open an image with openCV.

Programming

Opencv

Cpp

Cmake



Written by Sourabh Jigjinni

13 Followers

Follow



More from Sourabh Jigjinni

 Sourabh Jigjinni

How to export a Yolov7-tiny model via onnx to TensorRT on a Jetson Nano 4GB

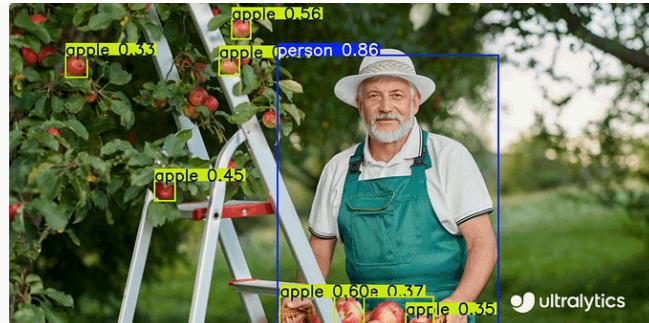
This article as of May 2023, is a (basic) guide, to help deploy a yolov7-tiny model to a Jetson nano 4GB.



May 8, 2023

[See all from Sourabh Jigjinni](#)

Recommended from Medium

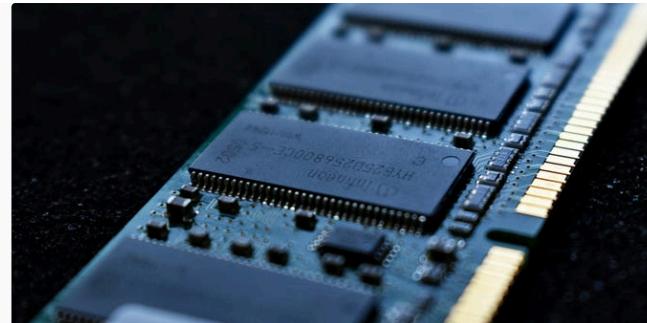


 Daniel García

Introducing YOLO11: The Next Leap in Computer Vision

Ultralytics has introduced YOLO11, the latest evolution in the YOLO (You Only Look Once)...

 4d ago  30



 Turyn Lim Banda

Embedded Software with STM32: DMA

An important concept in computer architecture

 Sep 11  1

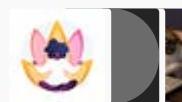


Lists



General Coding Knowledge

20 stories · 1607 saves



Stories to Help You Grow as a Software Developer

19 stories · 1385 saves



Coding & Development

11 stories · 825 saves



ChatGPT

21 stories · 816 saves

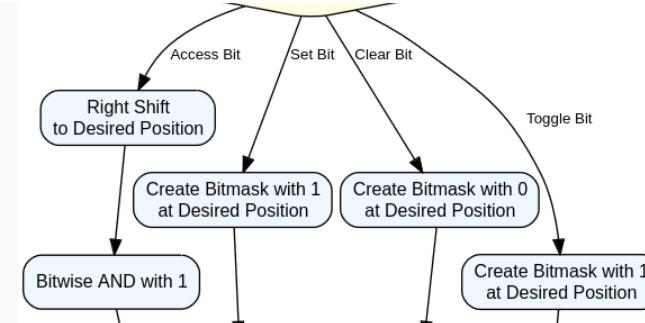


PointCloud-Slam-Image-We... in ROS Slam & Oth...

C ++ version receive serial gps rtk data, and then publish it as a ros...

The code implements a ROS node that receives serial data and forwards it into ROS...

★ May 1



Mohit Mishra in Nerd For Tech

Understanding Bit Manipulation in C

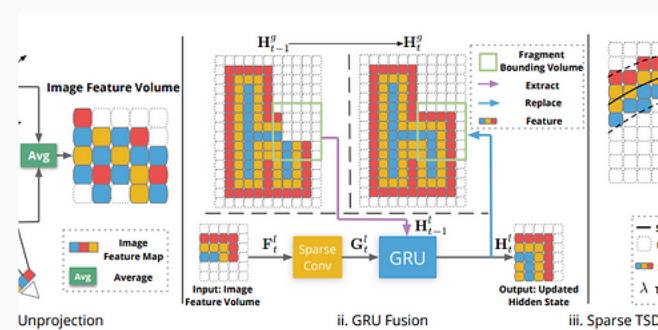
Accessing and Manipulating Individual Bits

★ Sep 23 69 1



Abhishek Jain

OpenCV basic functionalities in C++ (Morphological techniques)



The useless channel

Neural Recon: 4D reconstruction

In this blog, we are going to look at basic functionalities of openCV library in C++.

May 1  56



NeuralRecon: The latest method for real-time 3D scene reconstruction from a monocular...

Aug 20 



[See more recommendations](#)

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Terms](#) [Text to speech](#) [Teams](#)