

Building Allegro 5 Library using TDM-GCC and CMake

Step #	Slide #	Contents
1	2	Download and Install TDM MinGW
2	6	Test properly installed TDM-GCC version of MinGW
3	10	Download & unzip Allegro, DirectX, and Allegro Deps
4	15	Create and load Allegro 'build' and 'deps' folders
5	18	Install Cmake
6	22	Generate makefile – 32 bit build, monolithic, static
7	27	Make and install Allegro library
8	29	Turn in Screen Shots 1,2,3

Install TDM-GCC-MinGW Compiler

Download URL:

<https://jmeubank.github.io/tdm-gcc/download/>



Download a TDM-GCC installer:

[tdm-gcc-webdl.exe](#)

Minimal online installer. Select the components you want, and it downloads and unpacks them. Either edition, latest release only. (GCC 10.3.0)

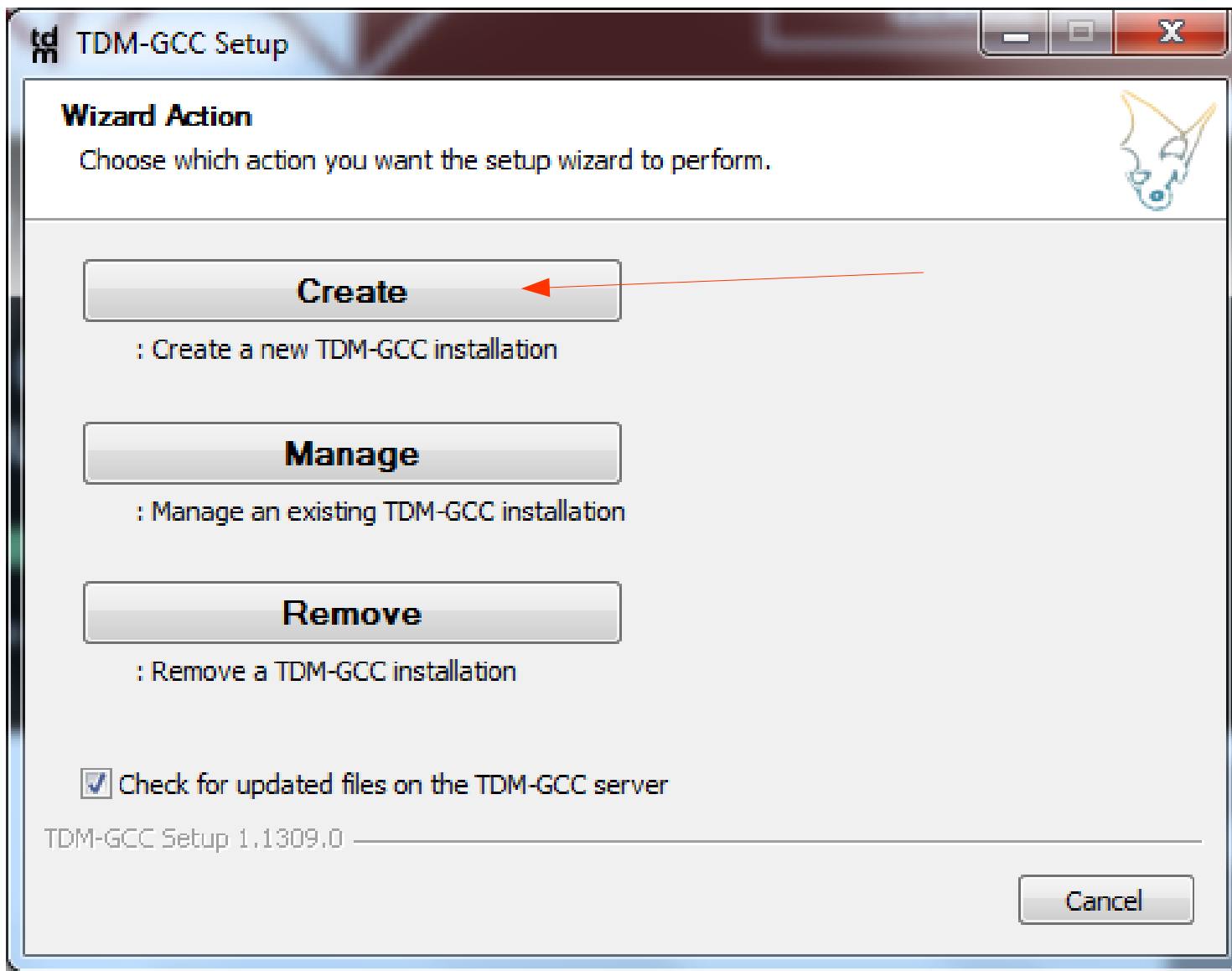
[tdm64-gcc-10.3.0-2.exe](#)

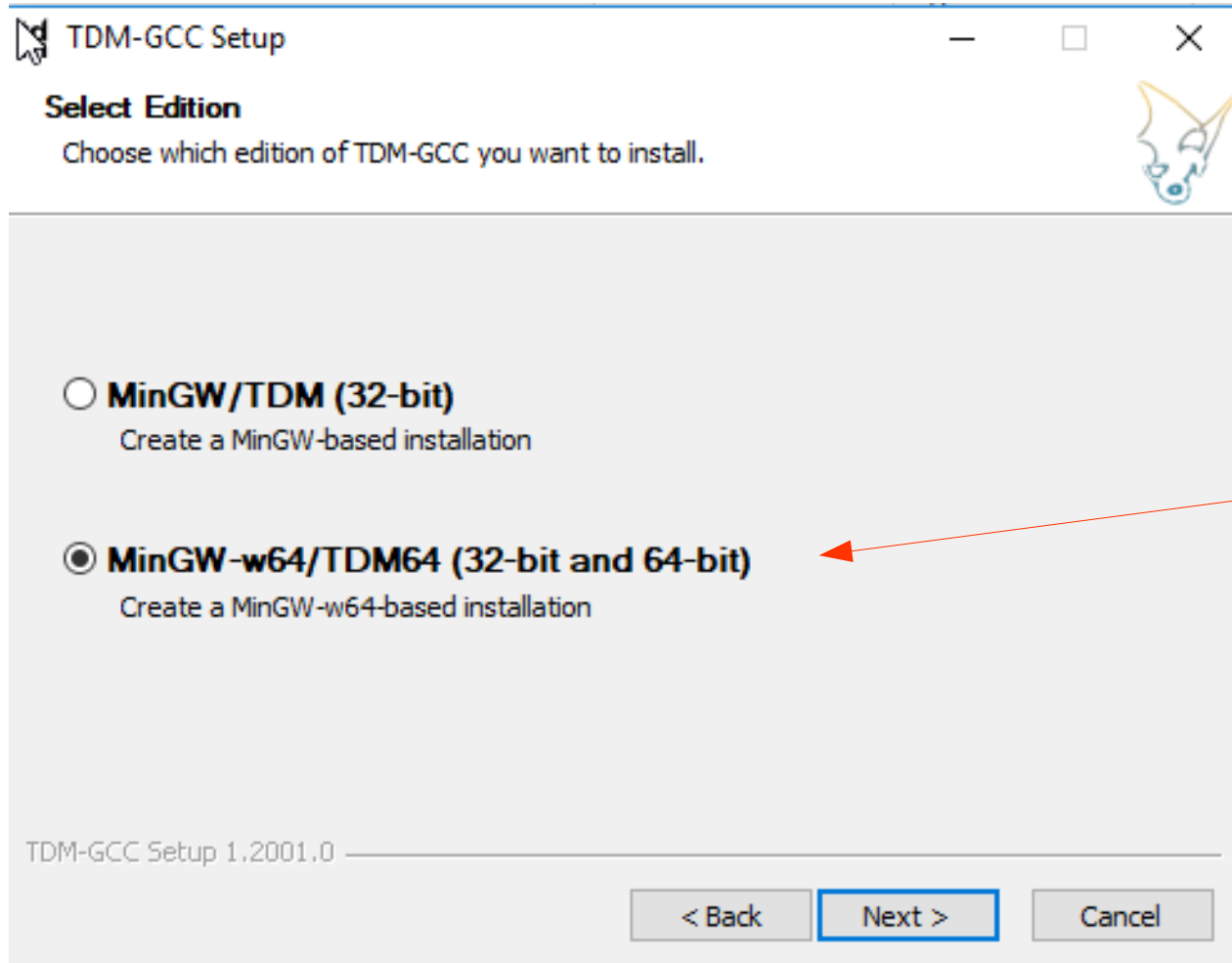
64+32-bit MinGW-w64 edition. Includes GCC C/C++, GNU binutils, mingw32-make, GDB (64-bit), the MinGW-w64 runtime libraries and tools, and the windows-default-manifest package.

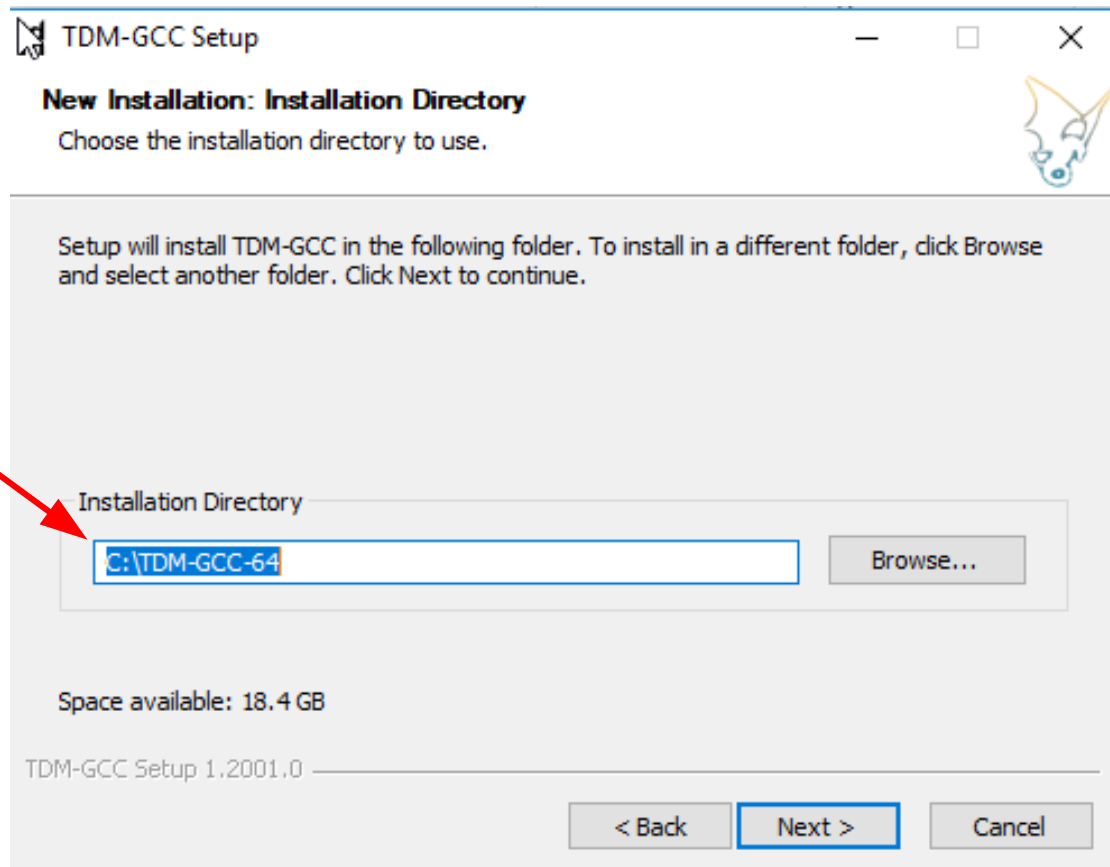
[tdm-gcc-10.3.0.exe](#)

32-bit-only MinGW.org edition. Includes GCC C/C++, GNU binutils, mingw32-make, GDB (32-bit), the MinGW.org mingwrt and w32api packages, and the windows-default-manifest package.

Download the
64+32 bit installer







Test it: Open a CMD prompt and enter 'gcc -v' to confirm correct installation. Response should be:

gcc tdm version

Troubleshoot:

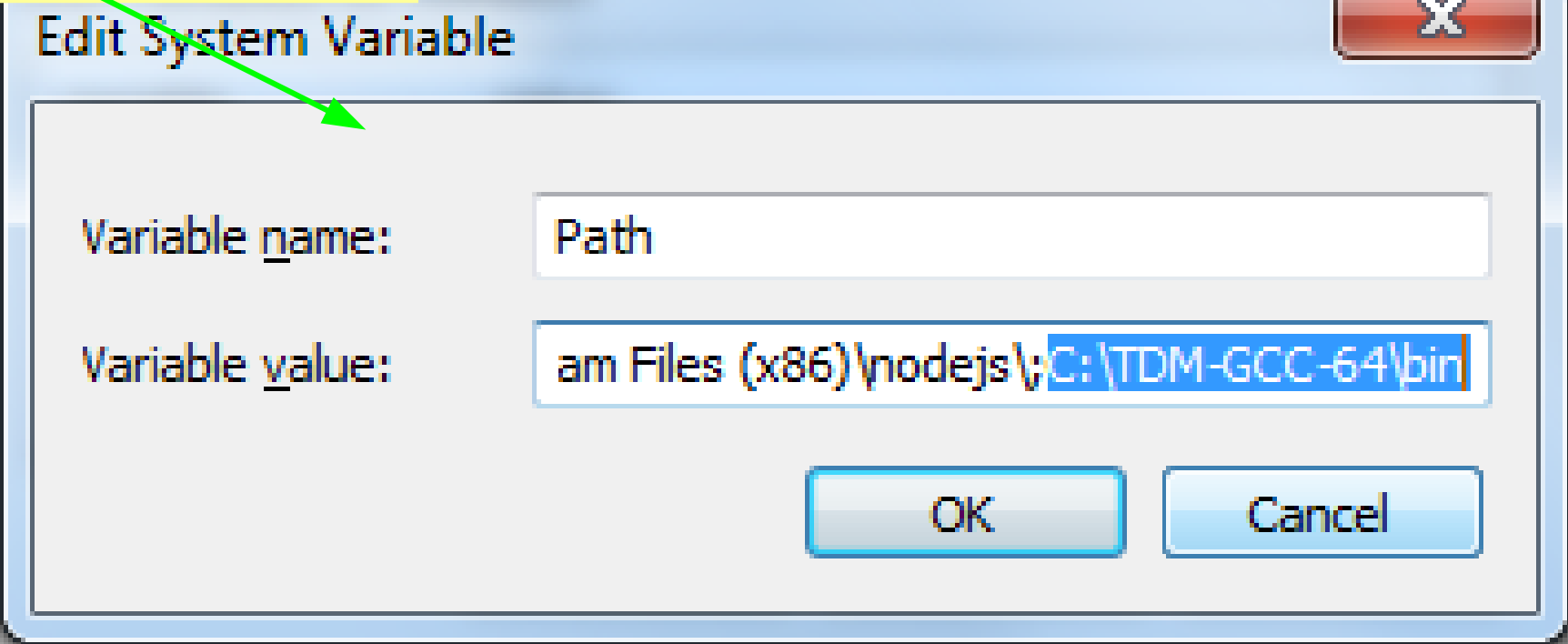
Make sure ;C:\TDM-GCC-64\bin is correct in the path statement
Make sure C:\MinGW\bin is not in the path statement

Next slide shows finding the Path variable

```
Command Prompt
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

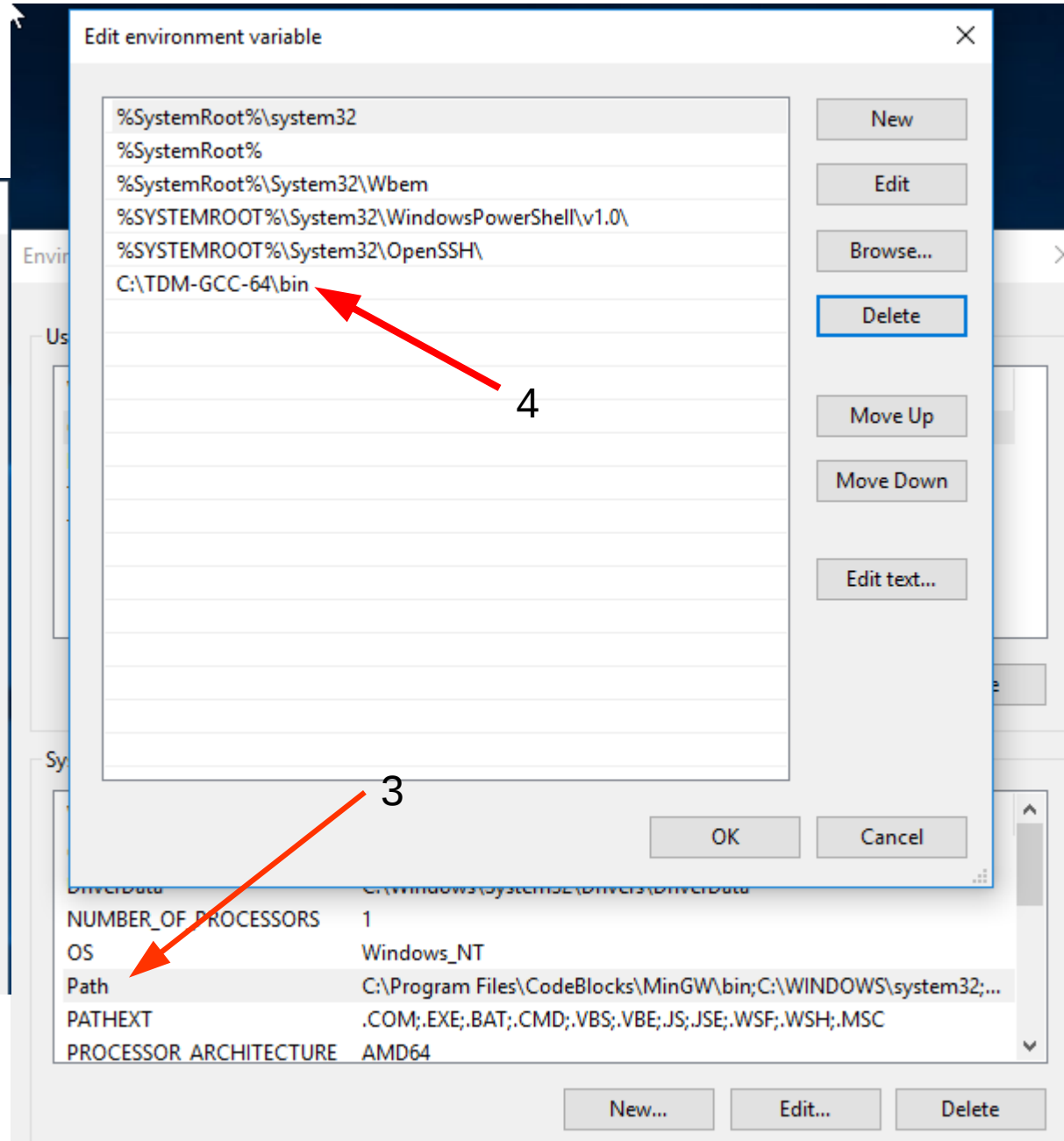
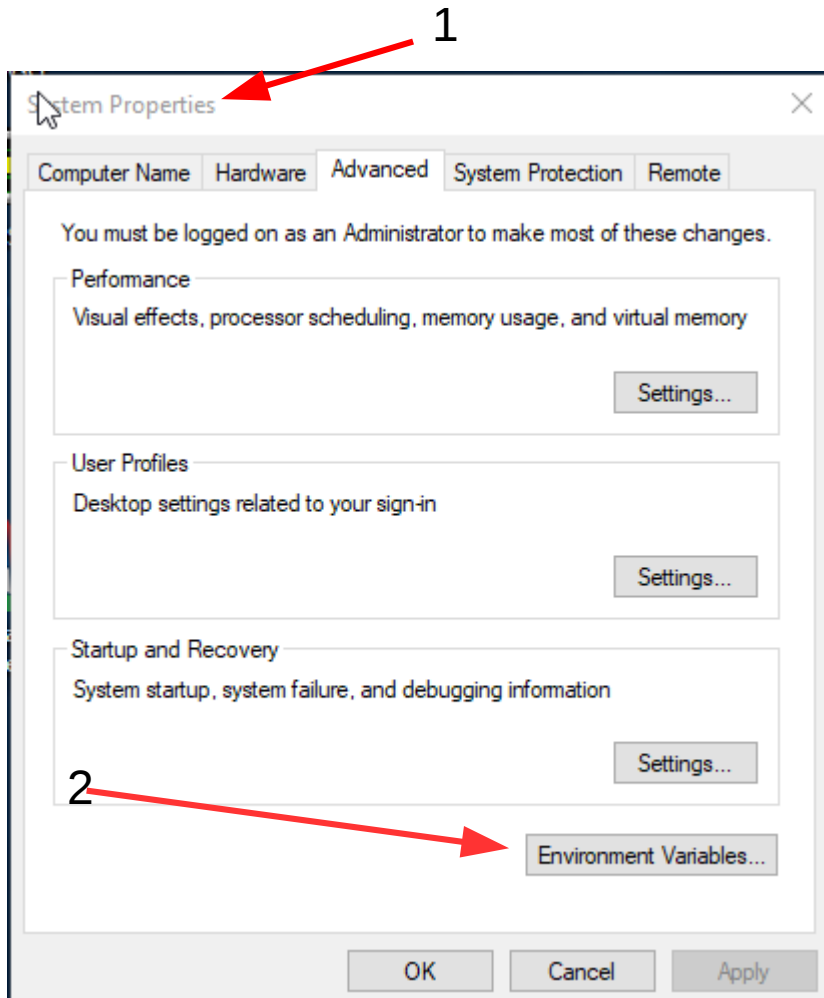
C:\Users\herman>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=C:/TDM-GCC-64/bin/./libexec/gcc/x86_64-w64-mingw32/9.2.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../src/gcc-git-9.2.0/configure --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --enable-languages=ada,c,c++,fortran,lto,objc,obj-c++ --enable-libgomp --enable-lto --enable-graphite --enable-shared --enable-threads=posix --disable-build-with-cxx --disable-build-poststage1-with-cxx --enable-libstdcxx-backtrace --enable-system-libs --enable-fully-dynamic-string --enable-libstdcxx-time --enable-ld --disable-werror --disable-nls --disable-win32-registry --enable-large-addresses --prefix=/mingw64tdm --with-local-prefix=/mingw64tdm --with-pkgversion=tdm64-1
Thread model: posix
gcc version 9.2.0 (tdm64-1)

C:\Users\herman>
```

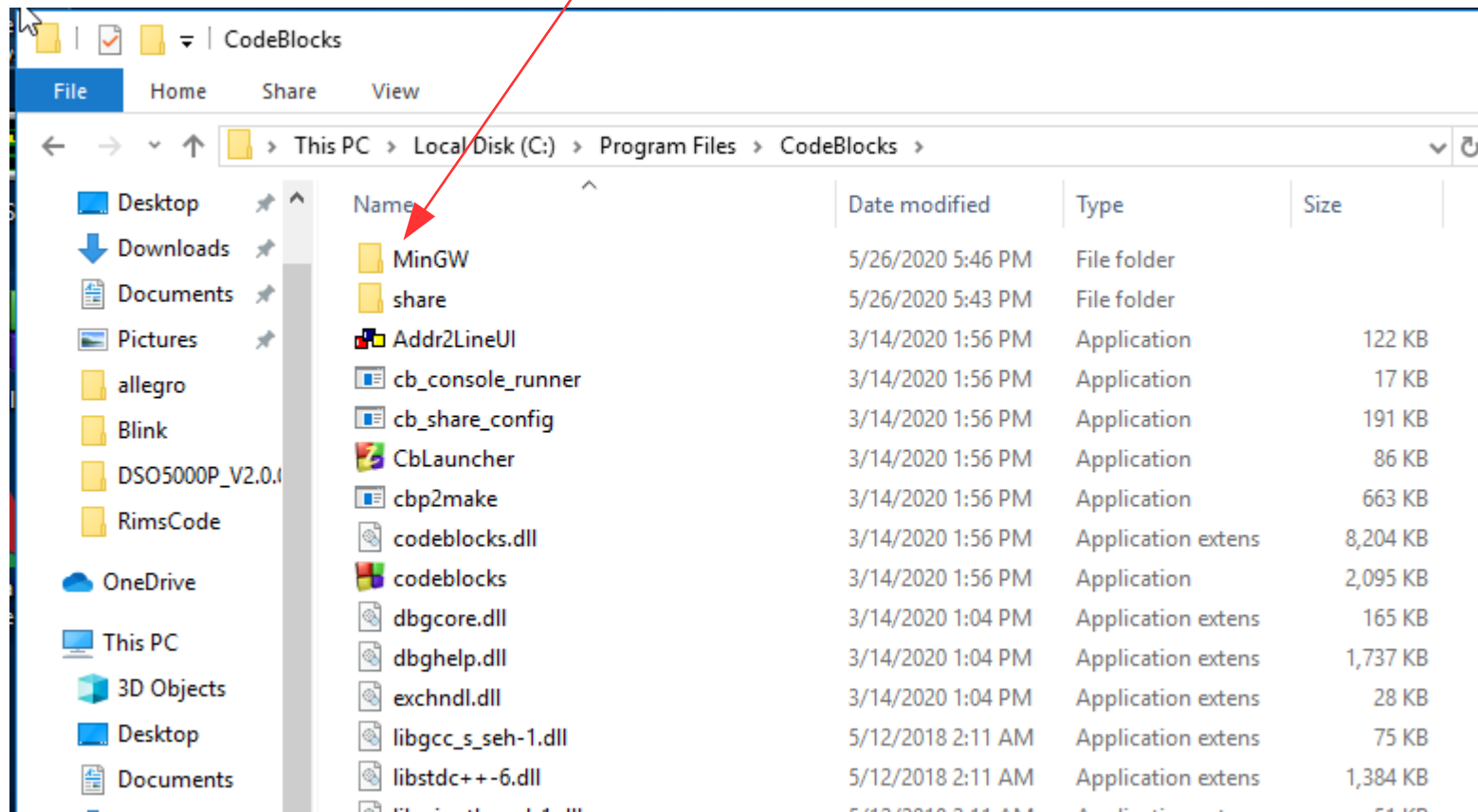


Troubleshoot only: Check the Windows

Path variable to not include
C:\MinGW\bin but TDM version



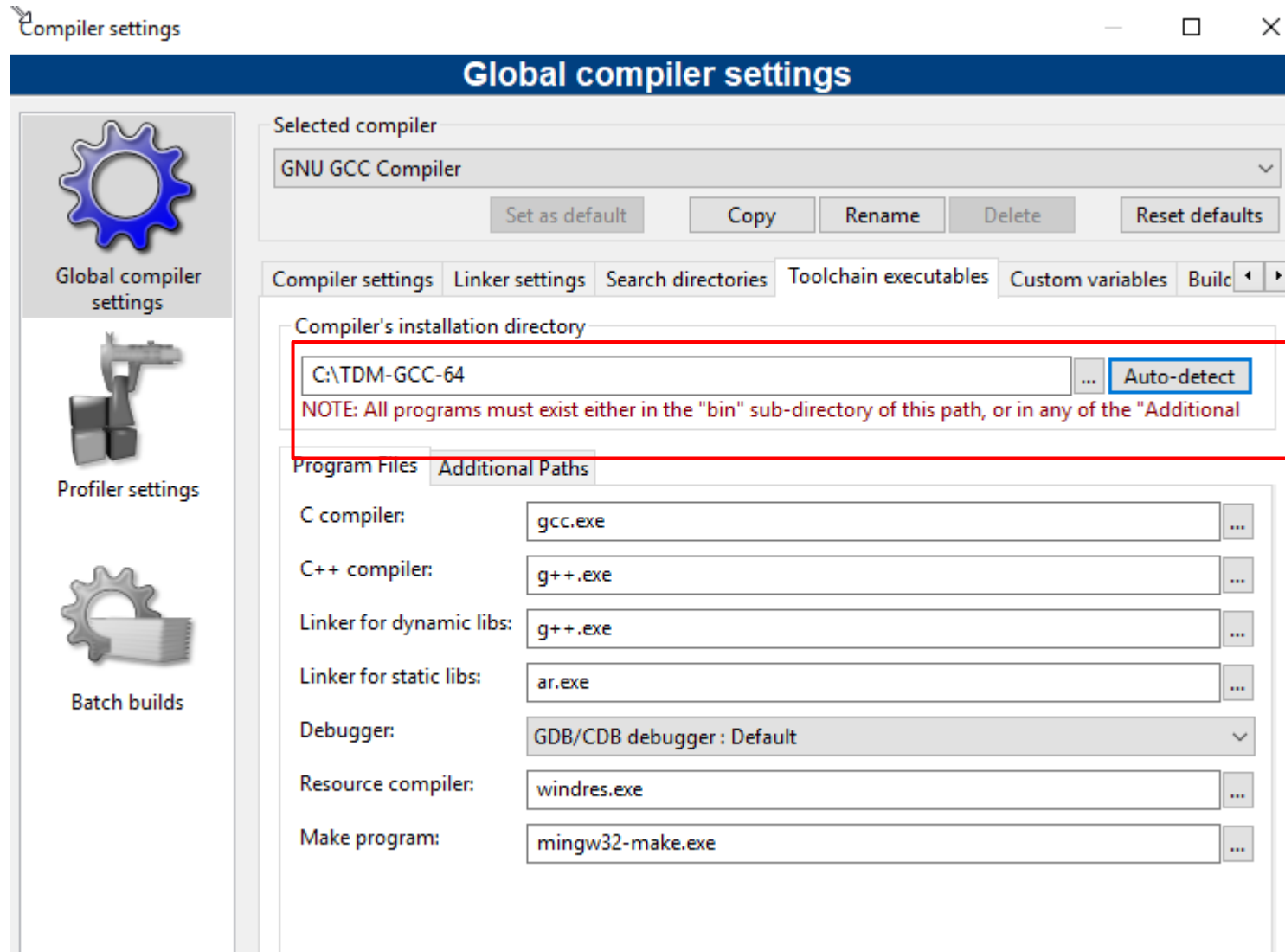
Clean up old stuff: If installed, in Code::Blocks folder DELETE MinGW folder or rename to MinGWBak to prevent conflicts



Load Code::Blocks

Go to Settings → Compiler → Toolchain executables

Make sure the TD-GCC-64 is the compiler detected. If not, back up and correct.





Install Allegro source files

A game programming library

- Allegro**
 - About
 - Git repository
 - License
 - Language bindings
- Downloads**
 - Latest version
 - Older versions
 - Extra addons
- Documentation**
 - Latest version
 - All versions
 - Tutorials
 - Books
 - Wiki
- Support**
 - Forums
 - Mailing lists
 - Bug tracker
 - IRC
- Community**
 - Allegro.cc
- Misc**
 - History
 - Older news
 - Logos
 - DIGMID
 - Links
 - Mirrors
 - Webmasters
 - Humor

Download

Download and unzip allegro-5.2. zip library from FIU archive:
<http://web.eng.fiu.edu/watsonh/EEL2880/Project/allegro-5.2.2.zip>

Here you can download releases of the Allegro 5 library.

- Source
- Windows
- MacOS X
- Linux
- iPhone
- Miscellaneous files

Instructions on how to build Allegro for each of the supported platforms are included in the source packages, either in the README files, or in the docs/build subdirectory.

Additional installation instructions can be found on the [wiki](#).

Download this copy

Allegro 5.2 source releases

Allegro 5.2 supports **Unix** (Linux, FreeBSD, etc.), **Windows**, **MacOS X** and **iPhone**. Allegro 5.2 is source compatible but not binary compatible with Allegro 5.0. You can read the [full change log](#).

Filename	Size	Description
allegro-5.2.2.zip	8.1M	Source code for all platforms in DOS/Windows friendly format.
allegro-5.2.2.tar.gz	7.2M	Source code for all platforms in Unix friendly format.
allegro-5.2.2.7z	6.1M	Source code for all platforms in DOS/Windows friendly format.
Browse all files		

Windows Binaries

The image shows a Windows File Explorer window with the 'Downloads' folder selected. The 'allegro-5.2.2' folder is highlighted in the file list. An 'Extract Compressed (Zipped) Folders' dialog box is open, showing the destination path 'C:\Users\herman\Downloads\allegro-5.2.2' in a text box. A red arrow points from the selected folder in the file list to the text box. The dialog box also has a 'Show extracted files when complete' checkbox checked and 'Extract' and 'Cancel' buttons at the bottom.

File Explorer Path: This PC > Downloads

File List:

- allegro_deps
- DSO5000P_Software
- dx9mgw
- energia-0101E0016-windows
- EZ430-UART
- ezFET-Lite
- TIUIFDriver
- wxDFS3X
- ZZBakallegro
- 7z1604-x64
- 83c8ec7ffd0a6ca8ac54b000du23fab2
- allegro_deps
- allegro-5.2.2
- anyconnect-win-4.5.02036-core-vpr
- c_robot_setup_1.0.130715
- ChromeSetup
- Cisco_WebEx_Add-On
- cmake-3.17.2-win64-x64
- codeblocks-20.03mingw-setup
- codeblocks-20.03-setup
- ConnectSetup

Dialog Box: Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\Users\herman\Downloads\allegro-5.2.2

Show extracted files when complete

Buttons: Extract, Cancel

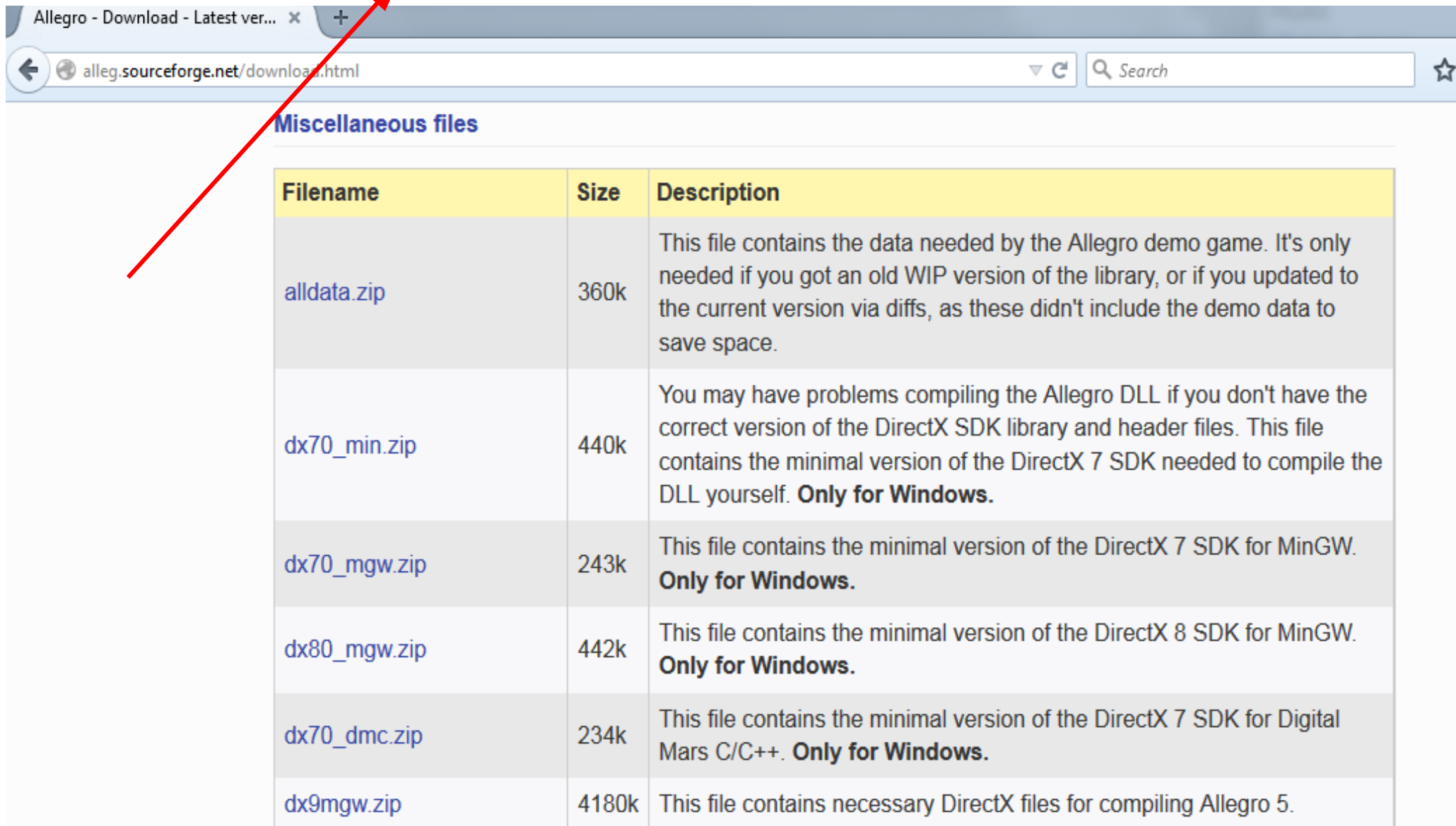
Name	Date Modified	Type	Size
allegro-5.2.2	5/26/2020 11:22 AM	Application	148,848 KB
codeblocks-20.03-setup	5/14/2020 11:24 AM	Application	36,574 KB
ConnectSetup	12/12/2017 11:19	Application	284 KB

Download and unzip dx9mgw.zip to Downloads folder

- direct download alternative link

<http://web.eng.fiu.edu/watsonh/EEL2880/project/dx9mgw.zip>

Slide 12



Allegro - Download - Latest ver... x +

alleg.sourceforge.net/download.html

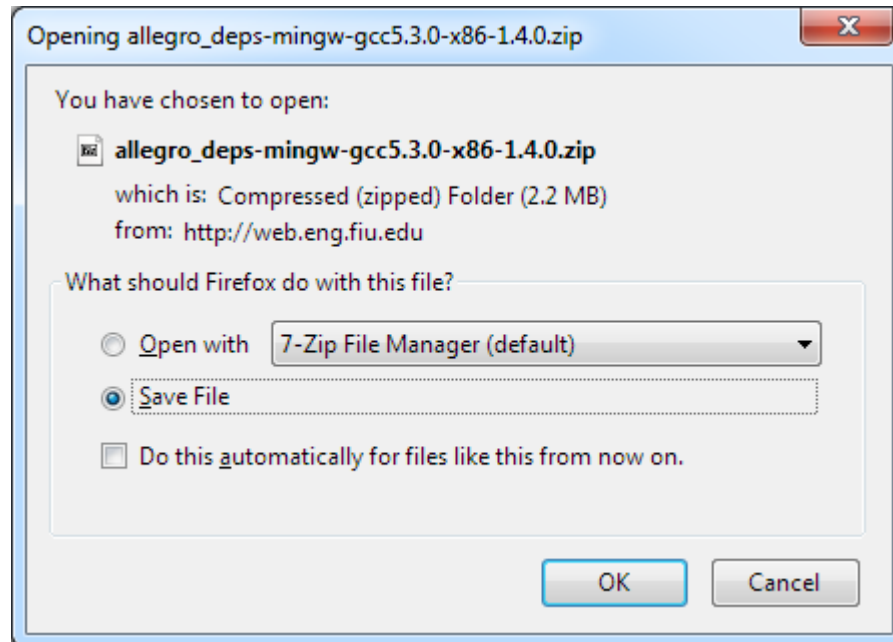
Miscellaneous files

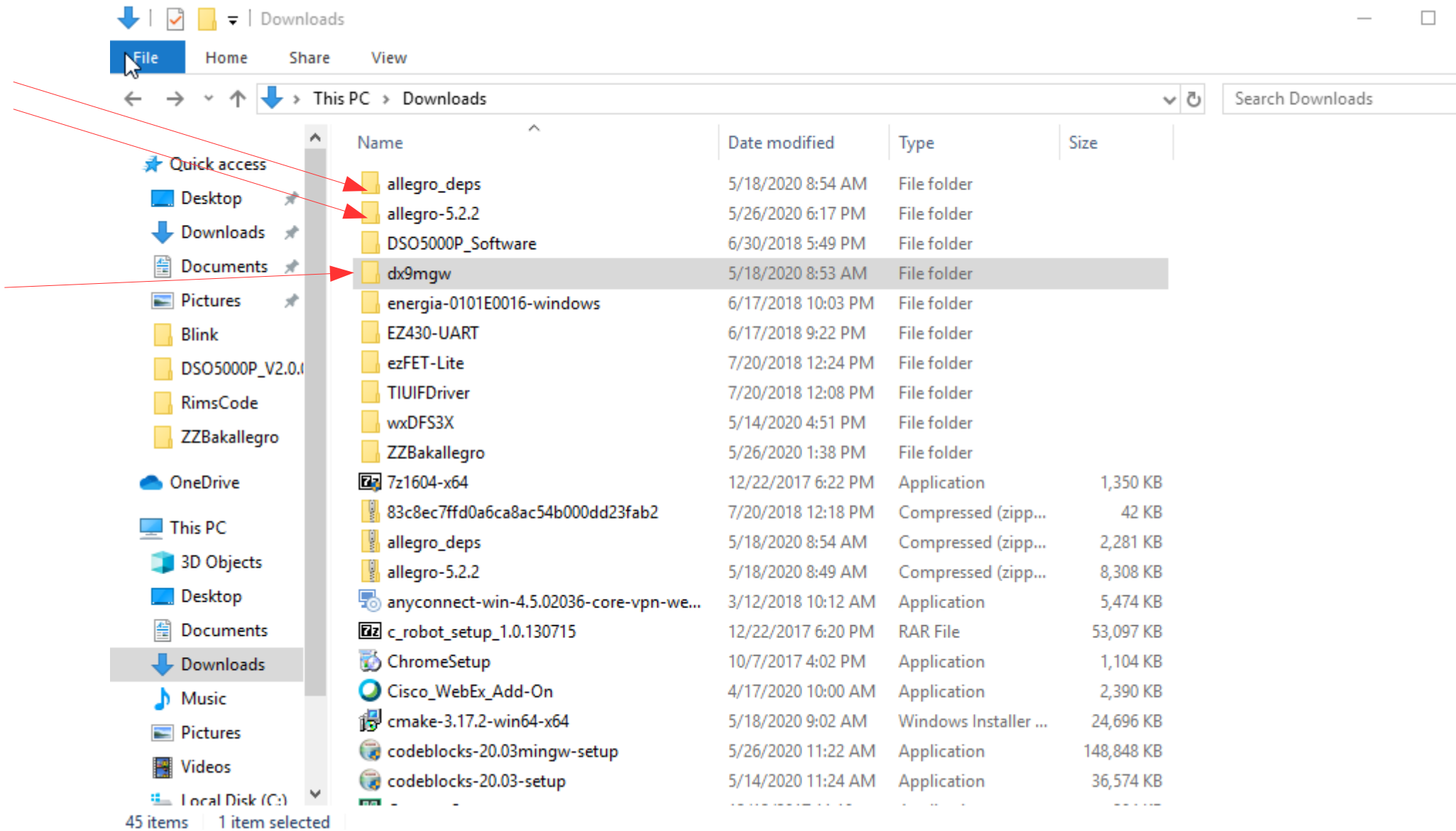
Filename	Size	Description
alldata.zip	360k	This file contains the data needed by the Allegro demo game. It's only needed if you got an old WIP version of the library, or if you updated to the current version via diffs, as these didn't include the demo data to save space.
dx70_min.zip	440k	You may have problems compiling the Allegro DLL if you don't have the correct version of the DirectX SDK library and header files. This file contains the minimal version of the DirectX 7 SDK needed to compile the DLL yourself. Only for Windows.
dx70_mgw.zip	243k	This file contains the minimal version of the DirectX 7 SDK for MinGW. Only for Windows.
dx80_mgw.zip	442k	This file contains the minimal version of the DirectX 8 SDK for MinGW. Only for Windows.
dx70_dmc.zip	234k	This file contains the minimal version of the DirectX 7 SDK for Digital Mars C/C++. Only for Windows.
dx9mgw.zip	4180k	This file contains necessary DirectX files for compiling Allegro 5.

NOTE: directX is pre-compiled as a 32-bit library which forces the allegro build as 32-bit library

Download and unzip Allegro_Deps folder in Downloads

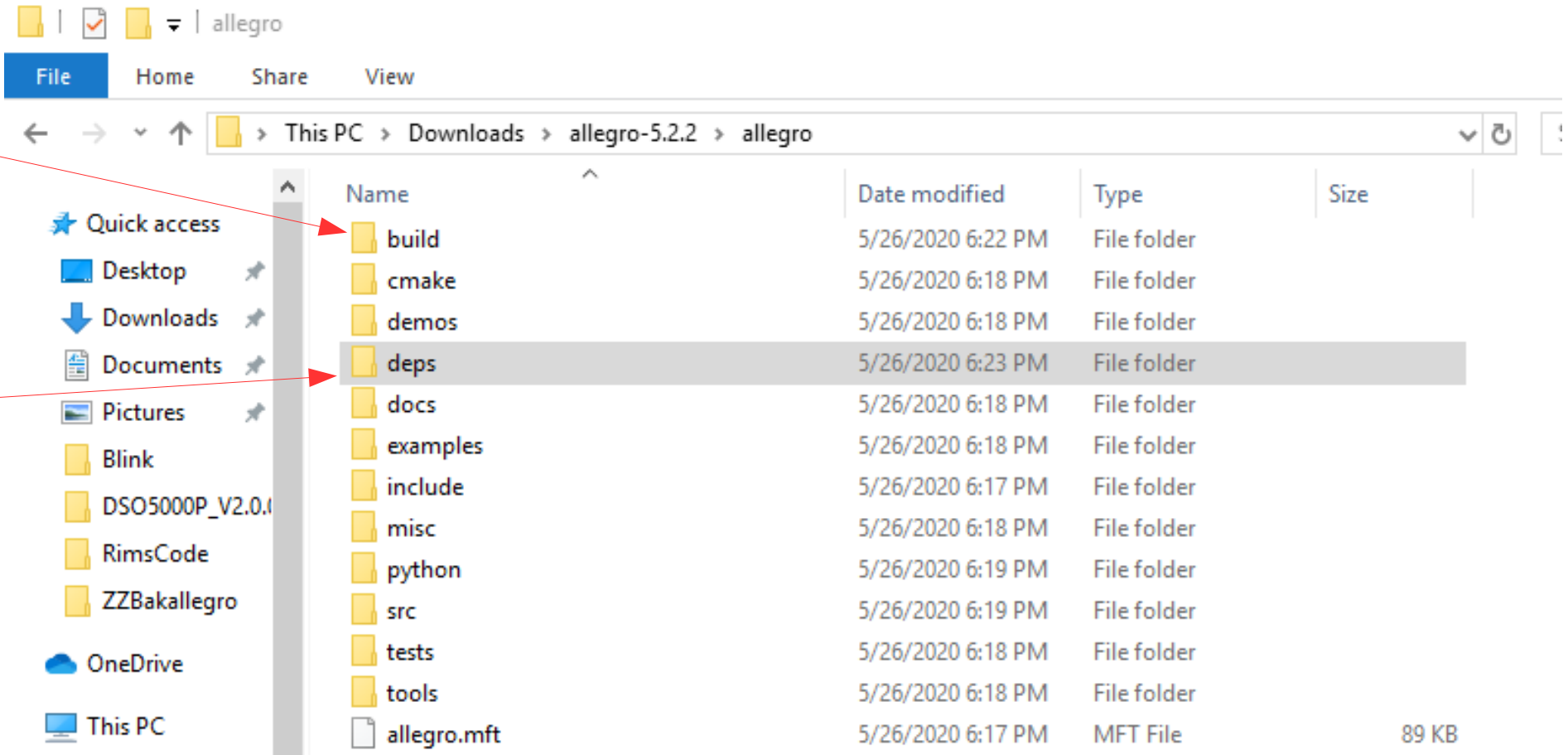
http://web.eng.fiu.edu/watsonh/EEL2880/Project/allegro_deps-mingw-gcc5.3.0-x86-1.4.0.zip

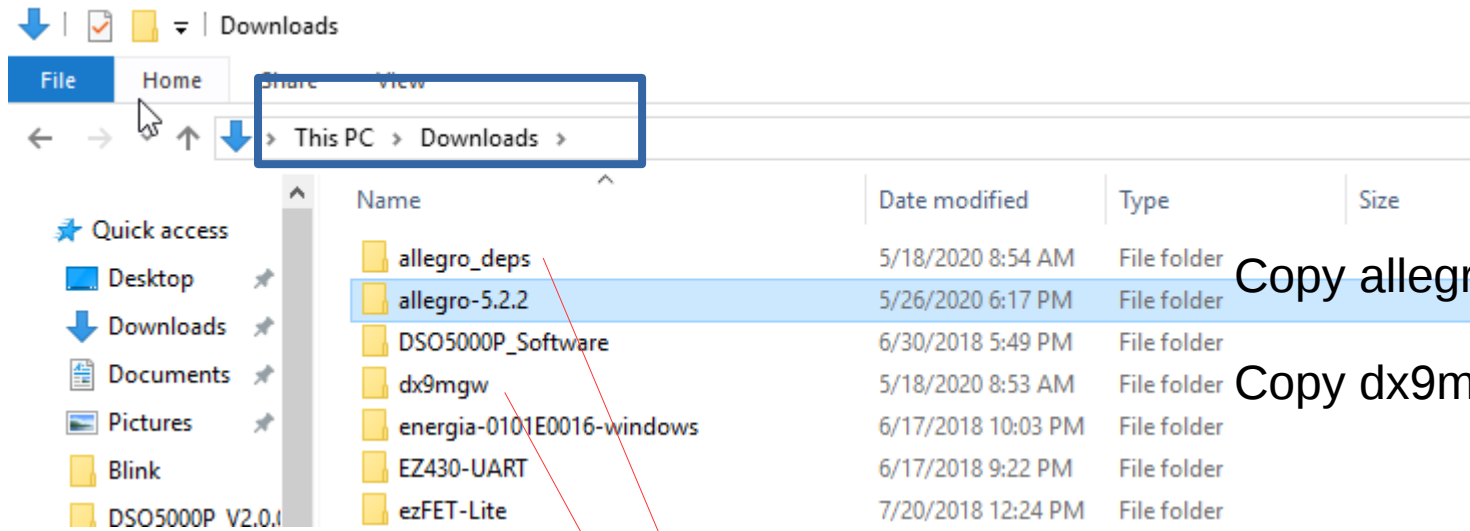




Inside the allegro folder, create two new folders:

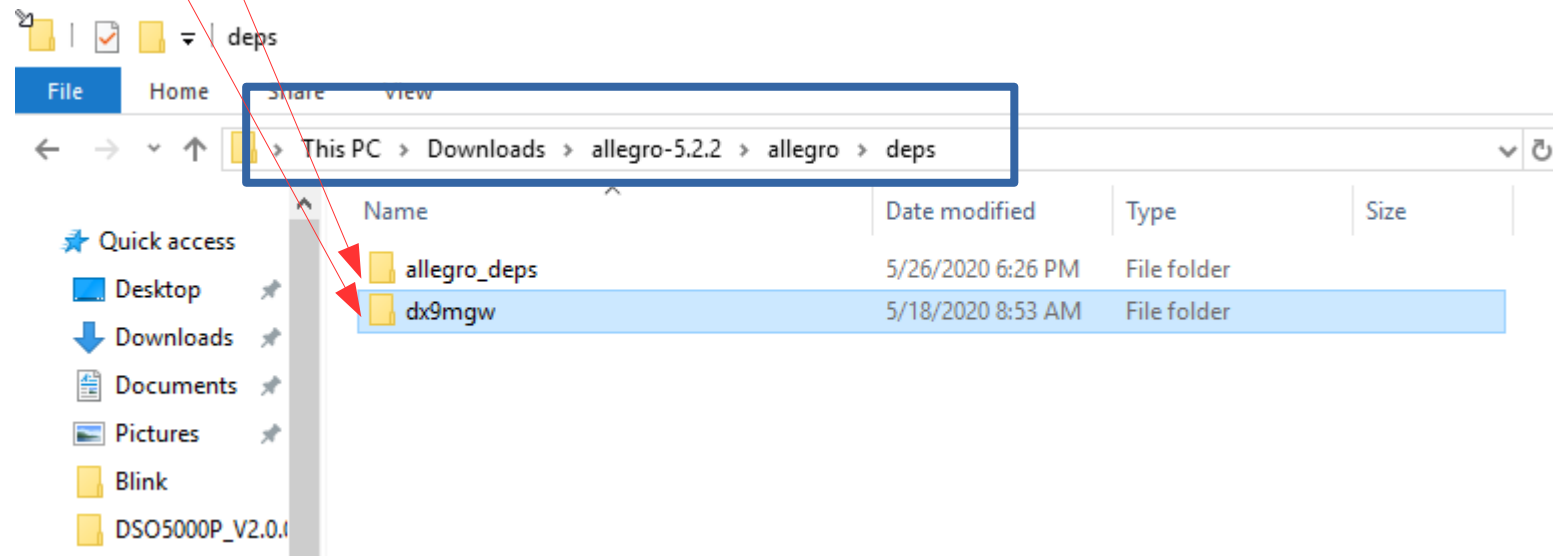
- 'build'
- 'deps'





Copy allegro_deps to deps folder

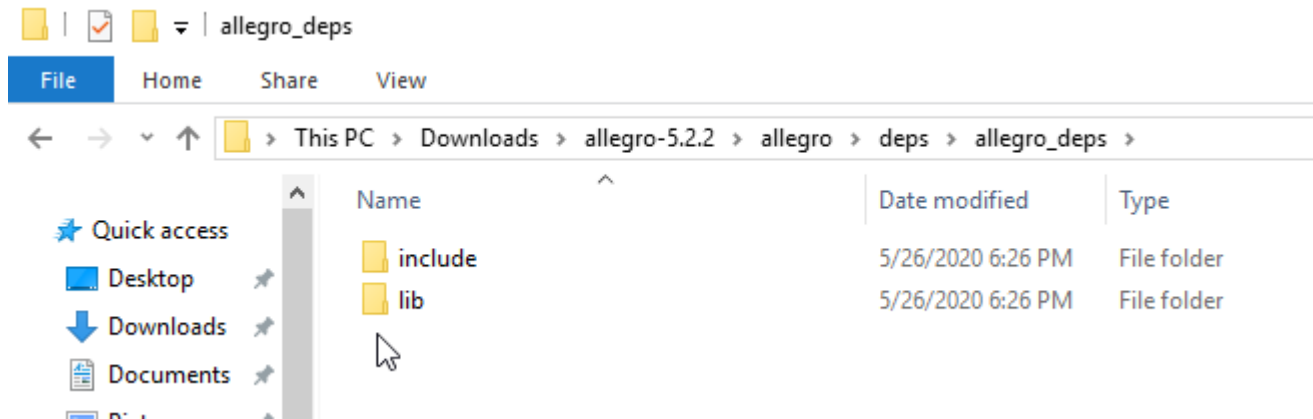
Copy dx9mgw to deps folder



Troubleshoot you may have to shuffle folders or files around:

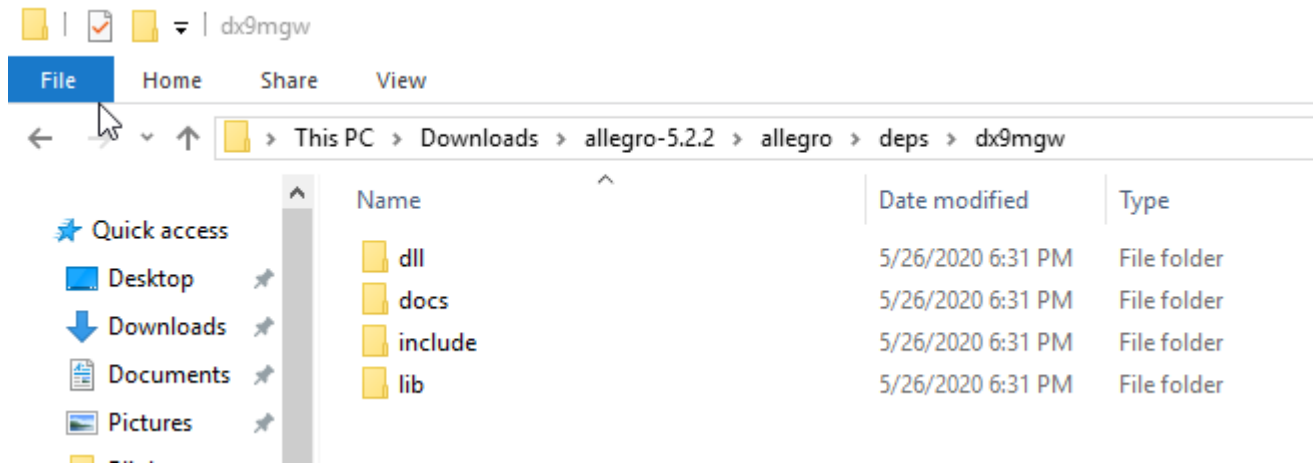
The inside of the allegro-deps folder should look like this

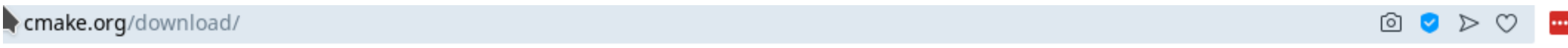
If not, you did not do something correct, back up and repeat to get this



The inside of the dx9mgw folder should look like this

If not, you did not do something correct, back up and repeat to get this





unpack them with zip or tar and follow the instructions in Readme.txt at the top of the source tree. See also the [CMake 3.17 Release Notes](#). Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.17.2.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.17.2.zip

Binary distributions:

Platform	Files
Windows win64-x64 Installer: Installer tool has changed. Uninstall CMake 3.4 or lower first!	cmake-3.17.2-win64-x64.msi

Windows win64-x64 ZIP [cmake-3.17.2-win64-x64.zip](#)

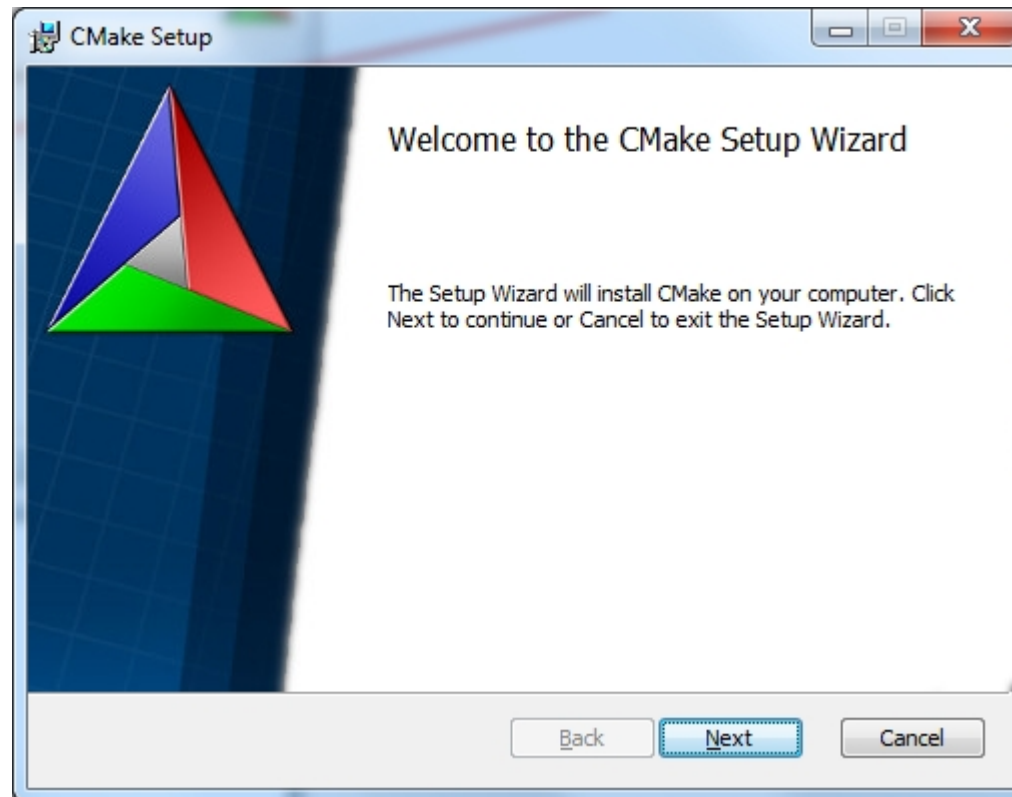
Windows win32-x86 Installer: **Installer tool has changed. Uninstall CMake 3.4 or lower first!** [cmake-3.17.2-win32-x86.msi](#)

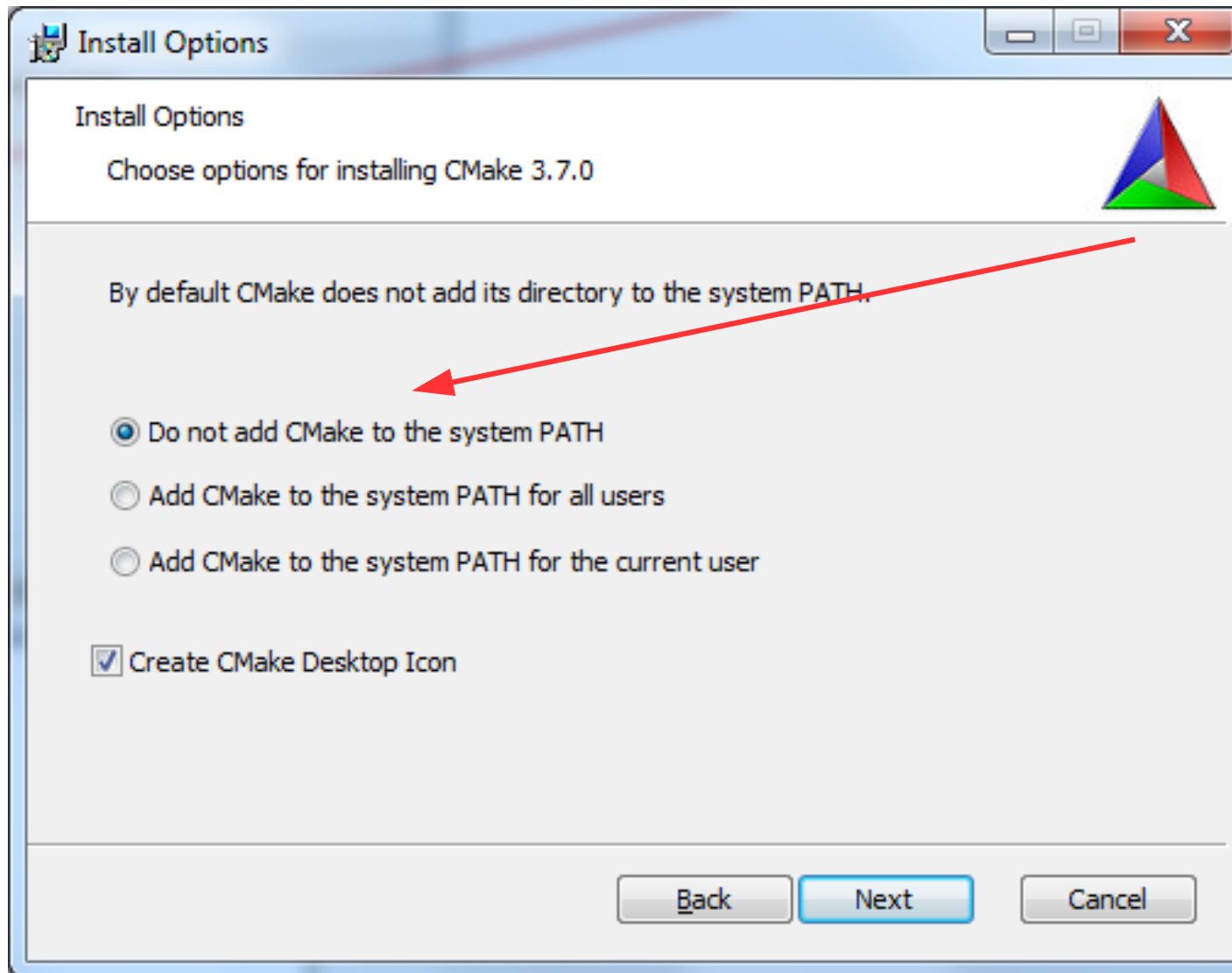
Windows win32-x86 ZIP [cmake-3.17.2-win32-x86.zip](#)

Mac OS X 10.7 or later [cmake-3.17.2-Darwin-x86_64.dmg](#)

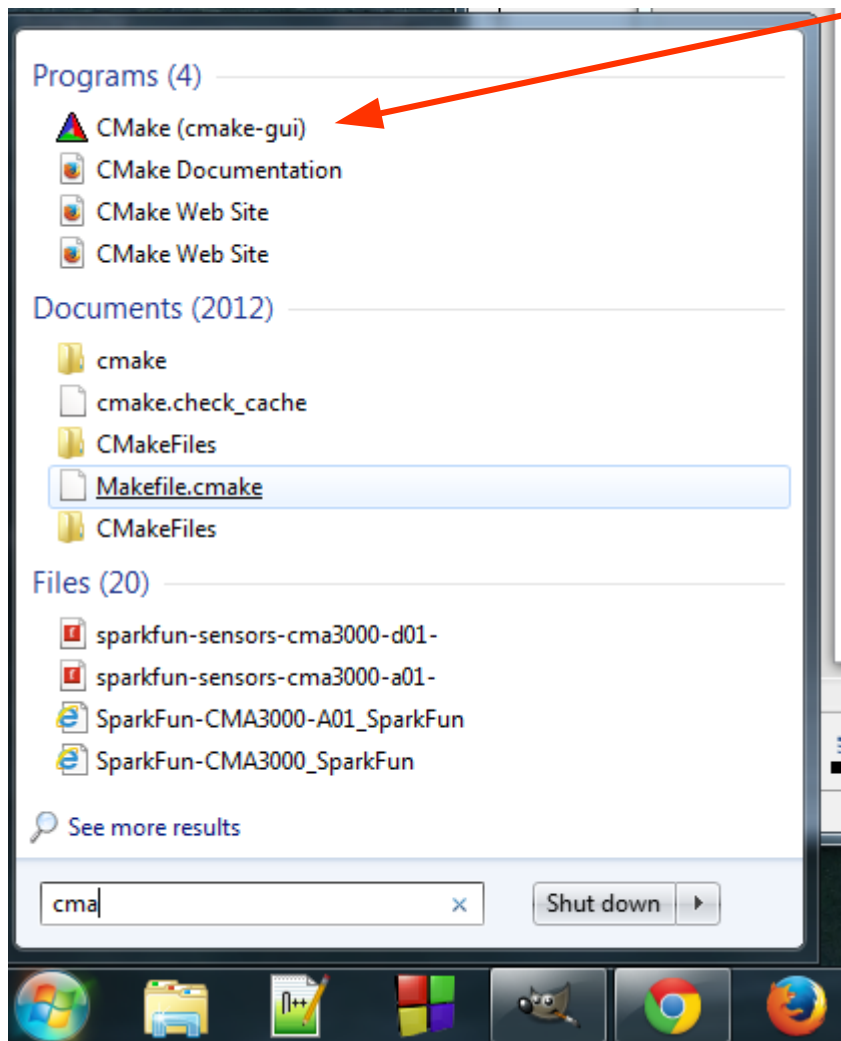
[cmake-3.17.2-Darwin-x86_64.tar.gz](#)

Download Cmake **Win64** installer and install on your computer



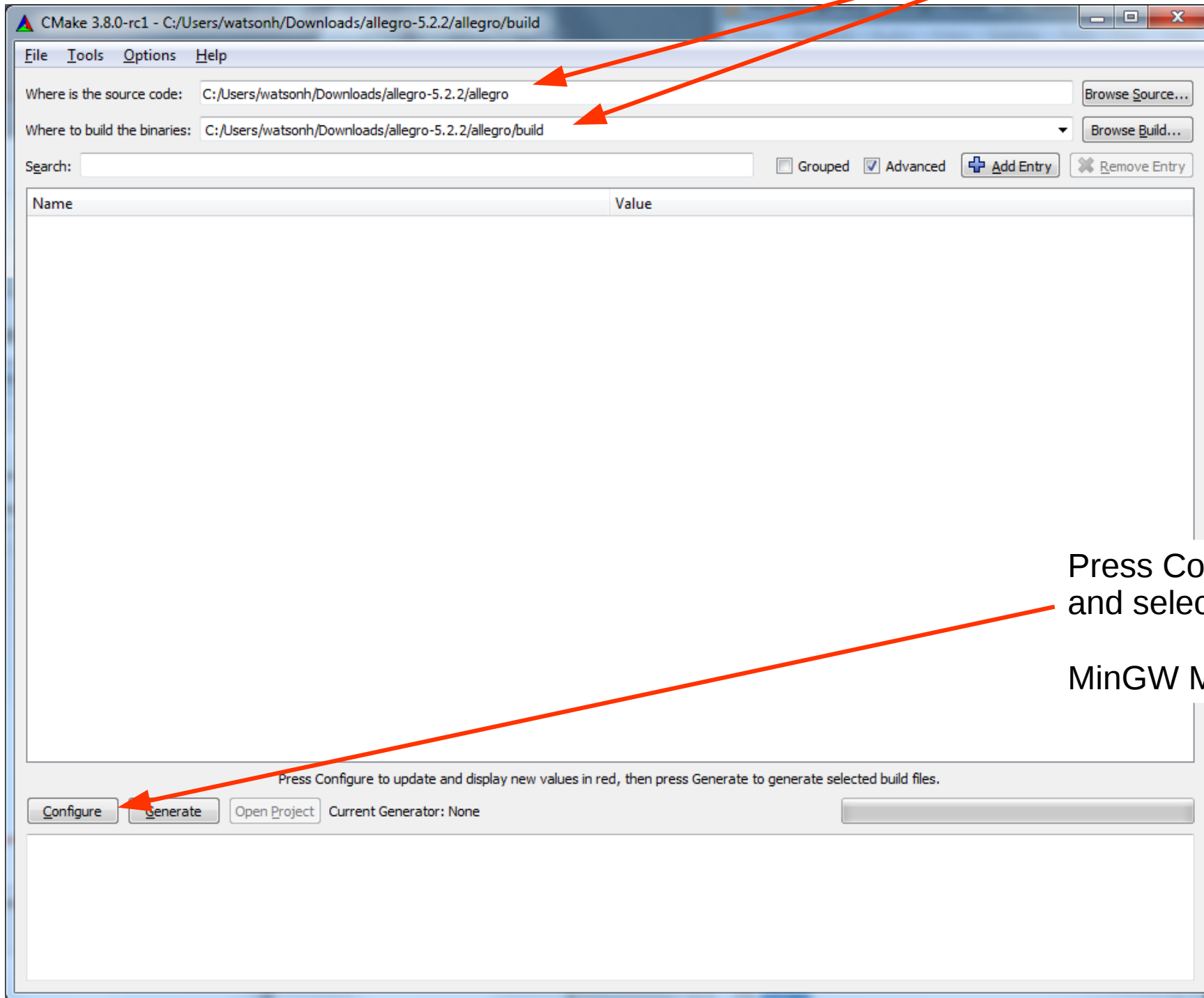


Start the Cmake-gui program



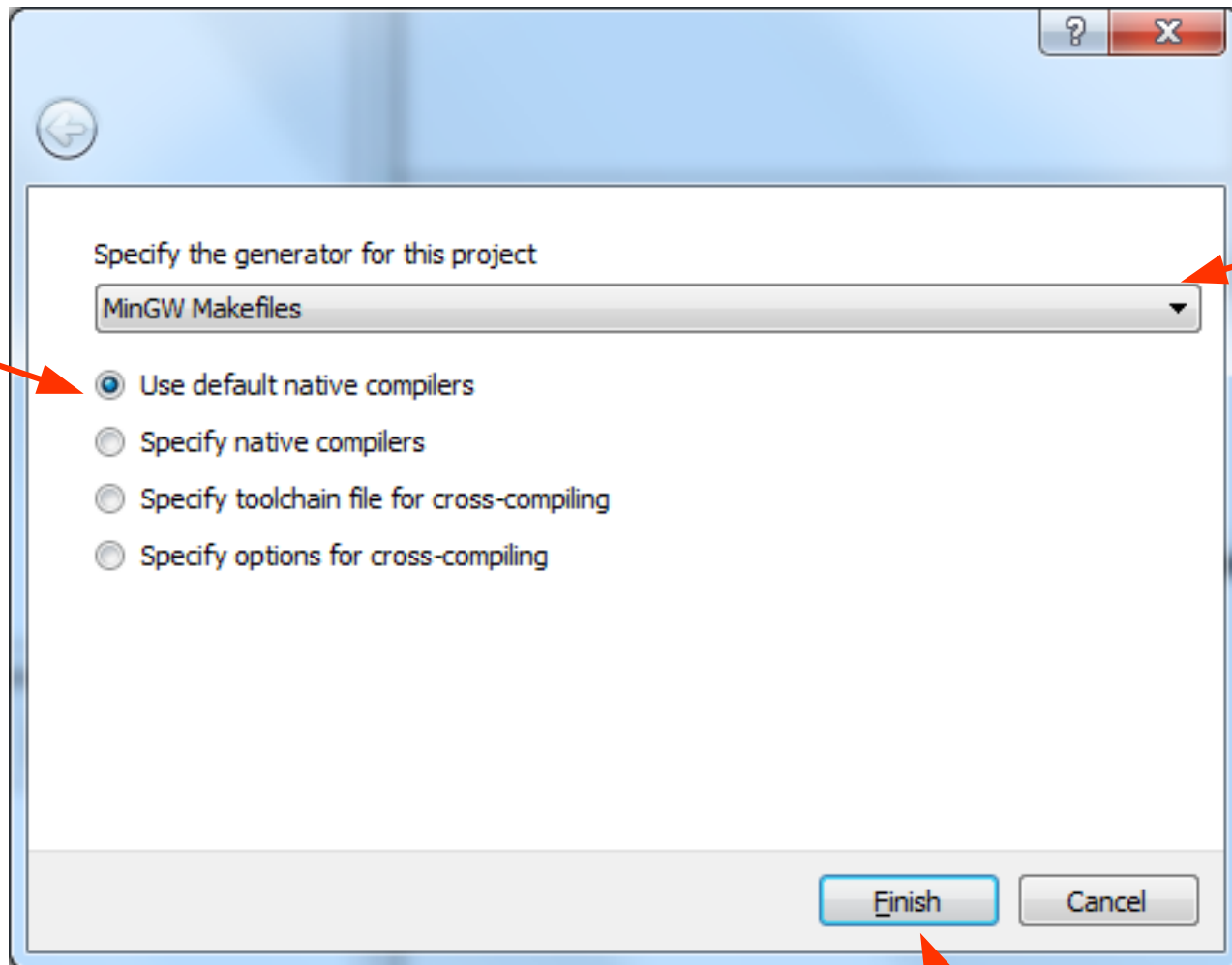
Enter the correct paths for the allegro source and the build folders

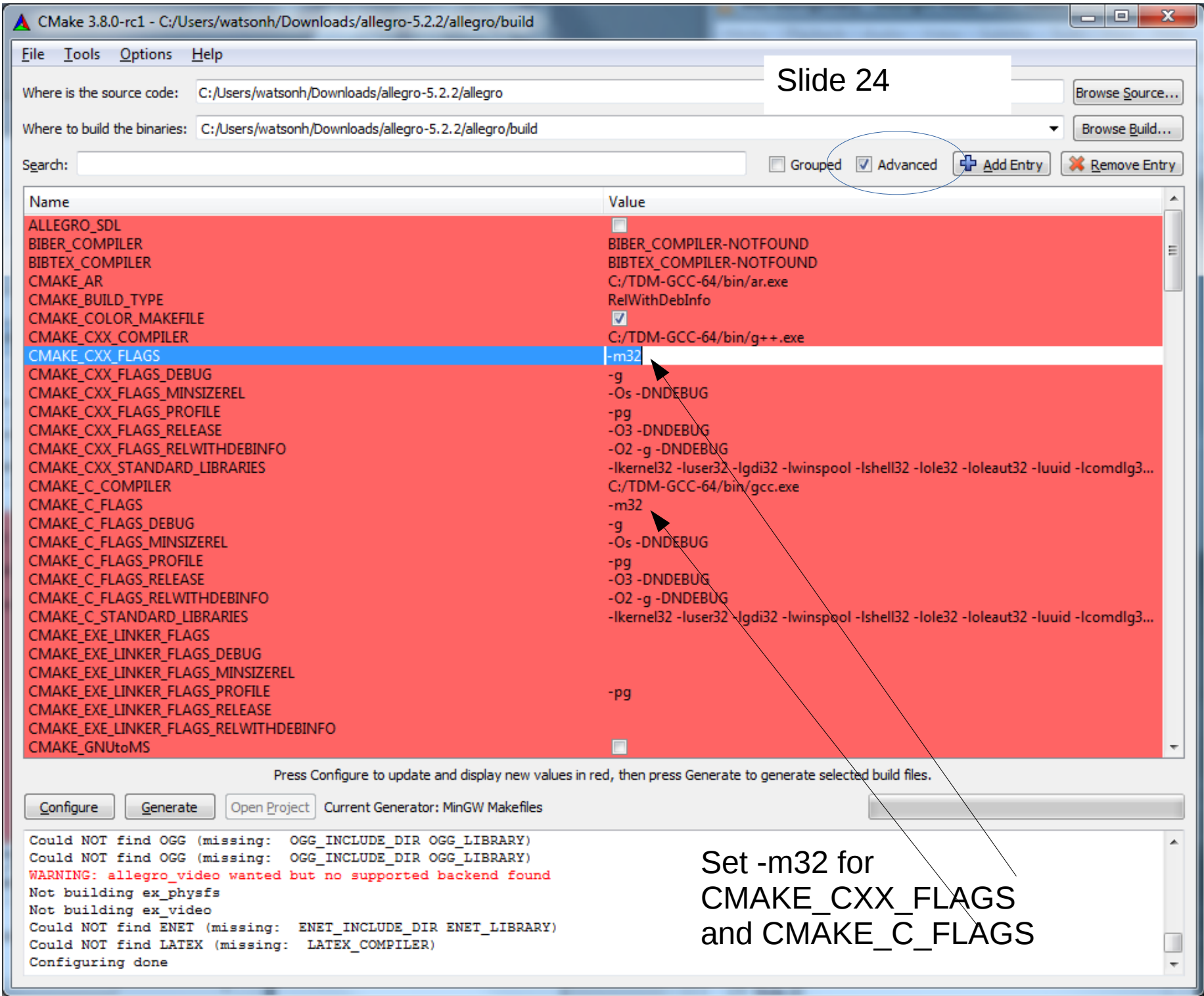
Slide 22



Press Configure button and select

MinGW Makefiles





Set -m32 for
CMAKE_CXX_FLAGS
and CMAKE_C_FLAGS

Where is the source code: C:/Users/Lenovo/Downloads/allegro

Where to build the binaries: C:/Users/Lenovo/Downloads/allegro/build

Search:

Grouped

Advanced

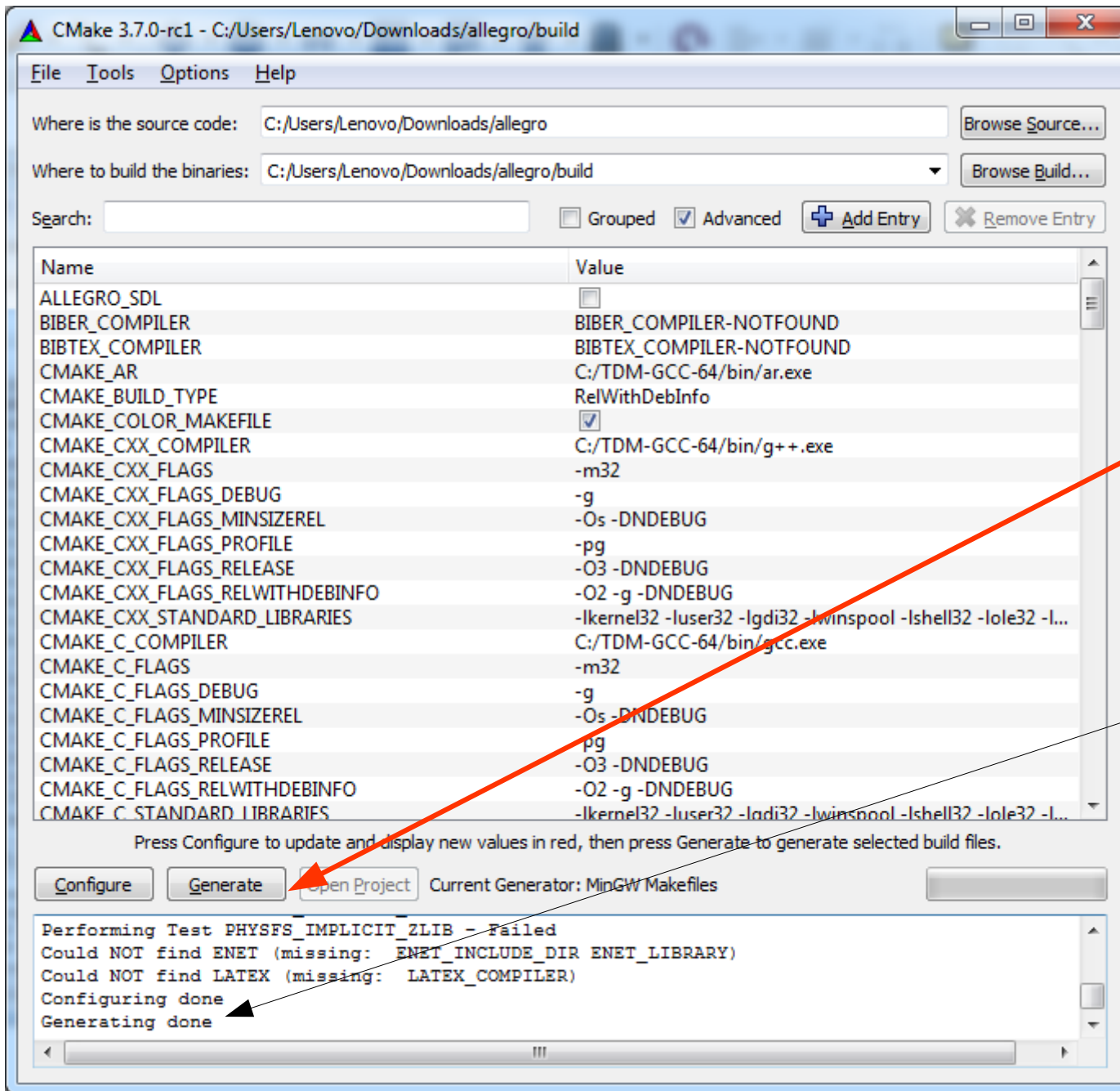
Name	Value
WANT_IMAGE	<input checked="" type="checkbox"/>
WANT_IMAGE_JPG	<input checked="" type="checkbox"/>
WANT_IMAGE_PNG	<input checked="" type="checkbox"/>
WANT_MEMFILE	<input checked="" type="checkbox"/>
WANT_MODAUDIO	<input checked="" type="checkbox"/>
WANT_MONOLITH	<input checked="" type="checkbox"/>
WANT_MUDFLAP	<input type="checkbox"/>
WANT_NATIVE_DIALOG	<input checked="" type="checkbox"/>
WANT_NATIVE_IMAGE_LOADER	<input checked="" type="checkbox"/>
WANT_OGG_VIDEO	<input checked="" type="checkbox"/>
WANT_OPENAL	<input checked="" type="checkbox"/>
WANT_OPENGL	<input checked="" type="checkbox"/>
WANT_OPENGL_S3TC_LOCKING	<input type="checkbox"/>
WANT_OPENSFL	<input checked="" type="checkbox"/>
WANT_OPUS	<input checked="" type="checkbox"/>
WANT_OSS	<input checked="" type="checkbox"/>
WANT_PHYSFS	<input checked="" type="checkbox"/>
WANT_POPUP_EXAMPLES	<input checked="" type="checkbox"/>
WANT_PRIMITIVES	<input checked="" type="checkbox"/>
WANT_PULSEAUDIO	<input checked="" type="checkbox"/>
WANT_PYTHON_WRAPPER	<input type="checkbox"/>
WANT_RELEASE_LOGGING	<input checked="" type="checkbox"/>
WANT_SHADERS_D3D	<input checked="" type="checkbox"/>
WANT_SHADERS_GL	<input checked="" type="checkbox"/>
WANT_STATIC_RUNTIME	<input checked="" type="checkbox"/>
WANT_TESTS	<input checked="" type="checkbox"/>
WANT_TREMOR	<input type="checkbox"/>

Scroll down and select WANT_MONOLITH

WANT_STATIC_RUNTIME

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: MinGW Makefiles



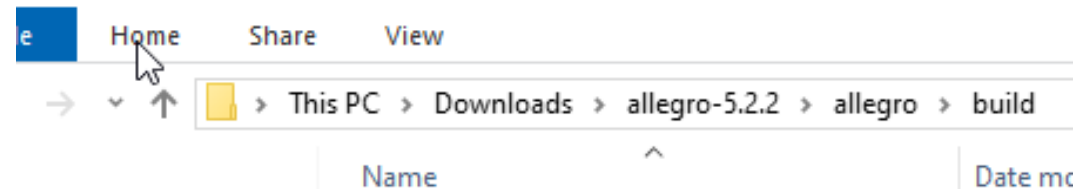
Press Generate button to create Makefile

Result screen:

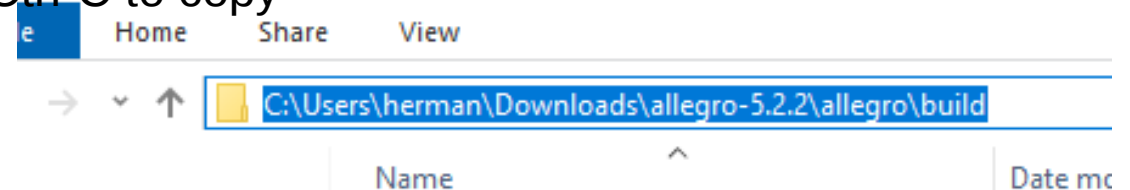
Note 'done' message

Build the Allegro Library

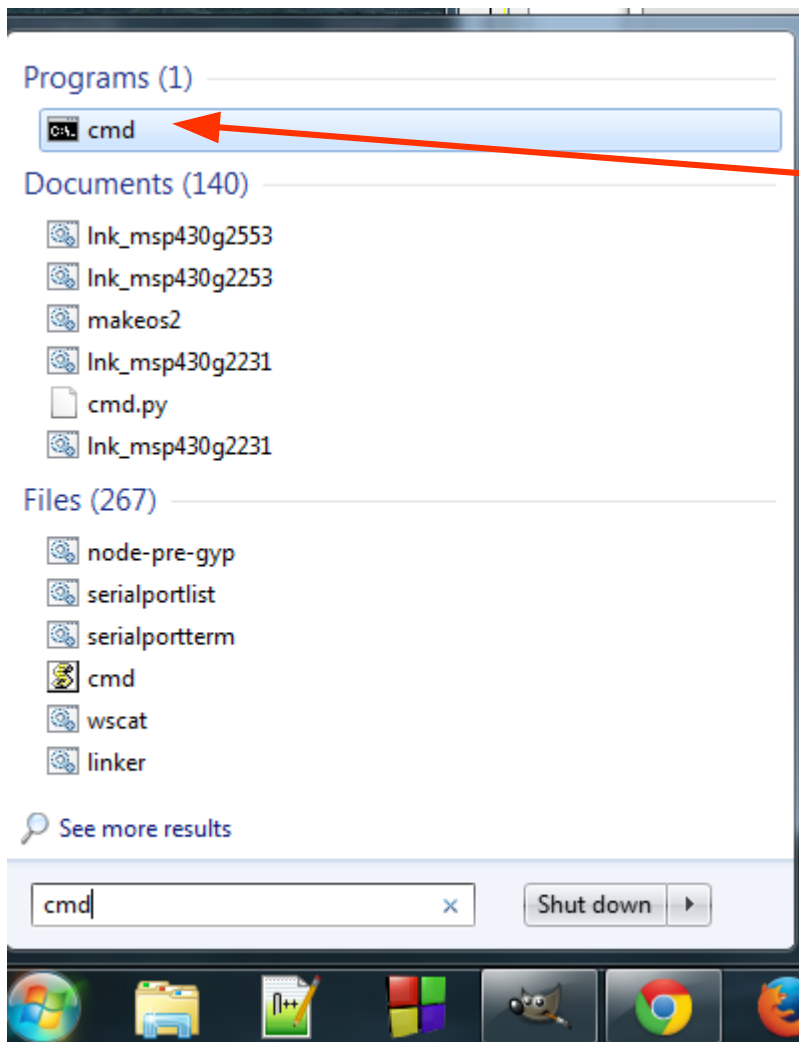
1. Open a command prompt
2. Open file explorer and go to 'build' folder



3. click on path bar – converts to full path
Ctrl-C to copy



4. enter 'cd ' in command prompt
then Right-Click and paste the full path – then
press enter to Change to the allegro\build folder
5. Type 'mingw32-make' command to
build the allegro library



```
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\herman>cd C:\Users\herman\Downloads\allegro-5.2.2\allegro\build
C:\Users\herman\Downloads\allegro-5.2.2\allegro\build>mingw32-make
```

```

C:\Users\watsonh\Downloads\allegro-5.2.2\allegro\build>mingw32-make install
[ 98%] Building C object examples/CMakeFiles/ex_resample_test.dir/ex_resample_test.c.obj
[ 98%] Linking C executable ex_resample_test.exe
[ 98%] Built target ex_resample_test
Scanning dependencies of target ex_saw
[ 98%] Building C object examples/CMakeFiles/ex_saw.dir/ex_saw.c.obj
[ 98%] Linking C executable ex_saw.exe
[ 98%] Built target ex_saw
Scanning dependencies of target ex_synth
[ 98%] Building CXX object examples/CMakeFiles/ex_synth.dir/ex_synth.cpp.obj
[ 98%] Building CXX object examples/CMakeFiles/ex_synth.dir/nihgui.cpp.obj
[ 98%] Linking CXX executable ex_synth.exe
[ 98%] Built target ex_synth
Scanning dependencies of target ex_menu
[ 98%] Building C object examples/CMakeFiles/ex_menu.dir/ex_menu.c.obj
[ 99%] Linking C executable ex_menu.exe
[ 99%] Built target ex_menu
Scanning dependencies of target test_driver
[ 99%] Building C object tests/CMakeFiles/test_driver.dir/test_driver.c.obj
[100%] Linking C executable test_driver.exe
[100%] Built target test_driver
Scanning dependencies of target docs
[100%] Built target docs

```

After build is complete,
enter install command:
'mingw32-make install'

This places library in compiler
system lib and bin folders

Install completed

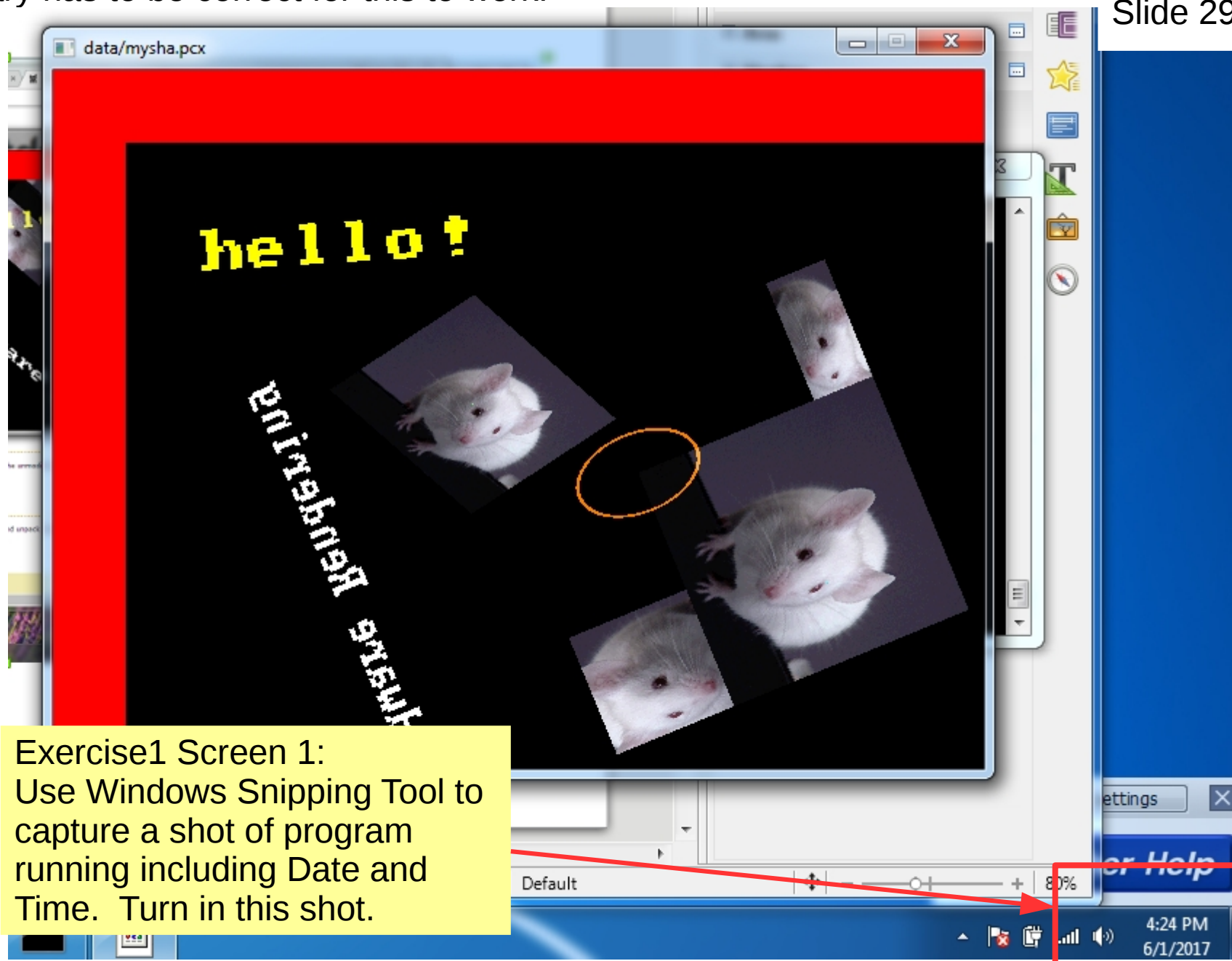
```

-- Installing: C:/TDM-GCC-64/include/allegro5/opengl/GLext/wgl_ext_list.h
-- Installing: C:/TDM-GCC-64/include/allegro5/platform/alplatf.h
-- Installing: C:/TDM-GCC-64/lib/liballegro_monolith.dll.a
-- Installing: C:/TDM-GCC-64/bin/allegro_monolith-5.2.dll
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_primitives.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_image.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_font.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_audio.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_acodec.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_ttf.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_color.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_memfile.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_physfs.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_native_dialog.h
-- Installing: C:/TDM-GCC-64/include/allegro5/allegro_video.h
C:\Users\herman\Downloads\allegro-5.2.2\allegro\build>

```

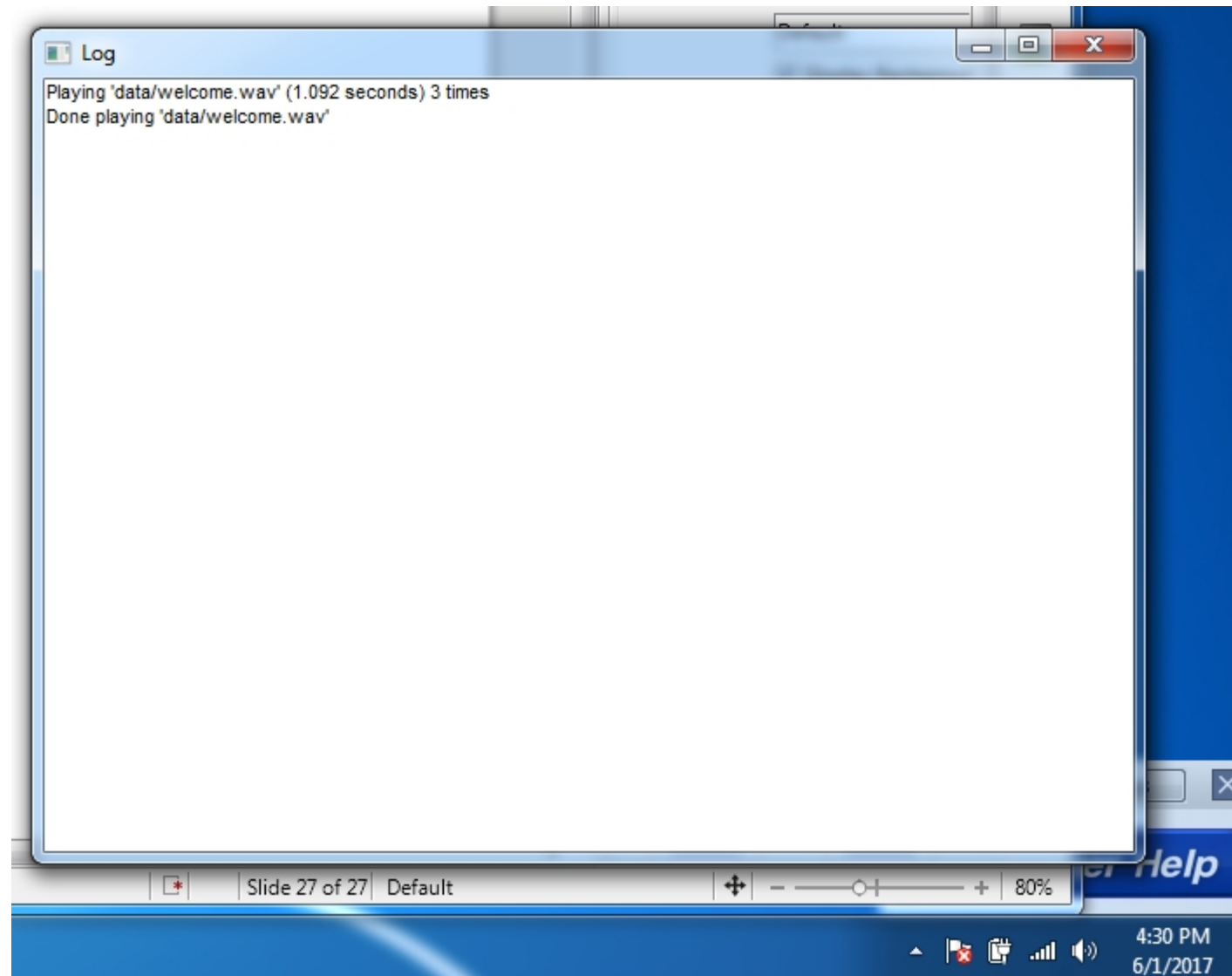
Exercise1 Screen 1 of 3: After Allegro build is complete, go to the 'build\examples' folder and run ex_transform.exe. To exit program, press 'Esc' button
If program does not work, check Slide 28 for troubleshooting steps and instructions. The library has to be correct for this to work.

Slide 29

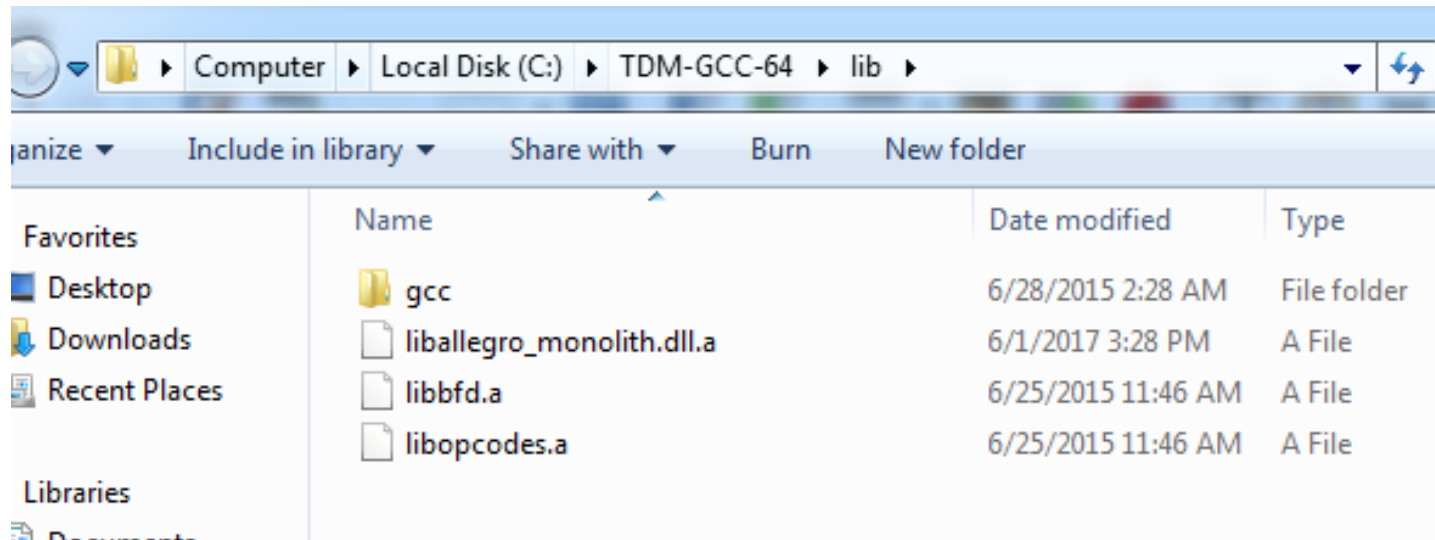


Exercise1 Screen 1:
Use Windows Snipping Tool to capture a shot of program running including Date and Time. Turn in this shot.

Exercise 1 Screen 2 of 3: In the same 'build\examples' folder and run ex_acodec.exe. Confirm that the audio is working. Again use Windows Snipping tool to capture an image of the Log screen plus the system Date and Time. This is Shot 2 of 3 to turn in.



Exercise 1 Screen 3 of 3: In Windows Explorer, go to the TDM-GCC-64\lib folder. Use Windows Snipping tool to capture an image of the folder screen. Confirm that the proper library was completed, 'liballegro_monolith.dll.a'. Turn in this image as Screen 3 of the submission document.



This library is the end result of the Allegro build process. If it is not in the folder then the build process was incorrect and did not follow the instructions exactly. Go back, review each step, and identify what was incorrect or missed.

This library is monolithic, and static. This means any '.exe' program built can be copied and run in any other folder or computer. Of course if the program requires images, text, data, or other files to run, those files and folders also have to be included in the same folder as the '.exe'.