

Bosch AutoVisionX

Team Name: Neutron Hive

Member 1 – Tushar Rathod

Member 2 – Aradhya Pitlawar

College – Walchand College of Engineering, Sangli

- **Guidelines and Steps to execute the program on Windows x64 systems**

1. Remove previous Mingw installations from system.
2. Delete MinGW installation directory (if necessary): If MinGW was installed manually or if there are any leftover files after uninstallation, you may need to manually delete the MinGW installation directory. The default installation directory for MinGW is usually C:\MinGW or C:\MinGW-w64. You can delete this directory and its contents using File Explorer or by running the following command in Command Prompt or PowerShell:

```
rm C:\MinGW
```

Note: Replace C:\MinGW with the actual installation directory if it's different.

```
PS C:\Users\ARADHYA> rm C:\MinGW
Confirm
The item at C:\MinGW has children and the Recurse parameter was not specified. If you continue, all children will be removed with the item. Are you sure you want to continue?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"):
```

Confirm your uninstallation by running

```
g++ --version
```

You should get the below output

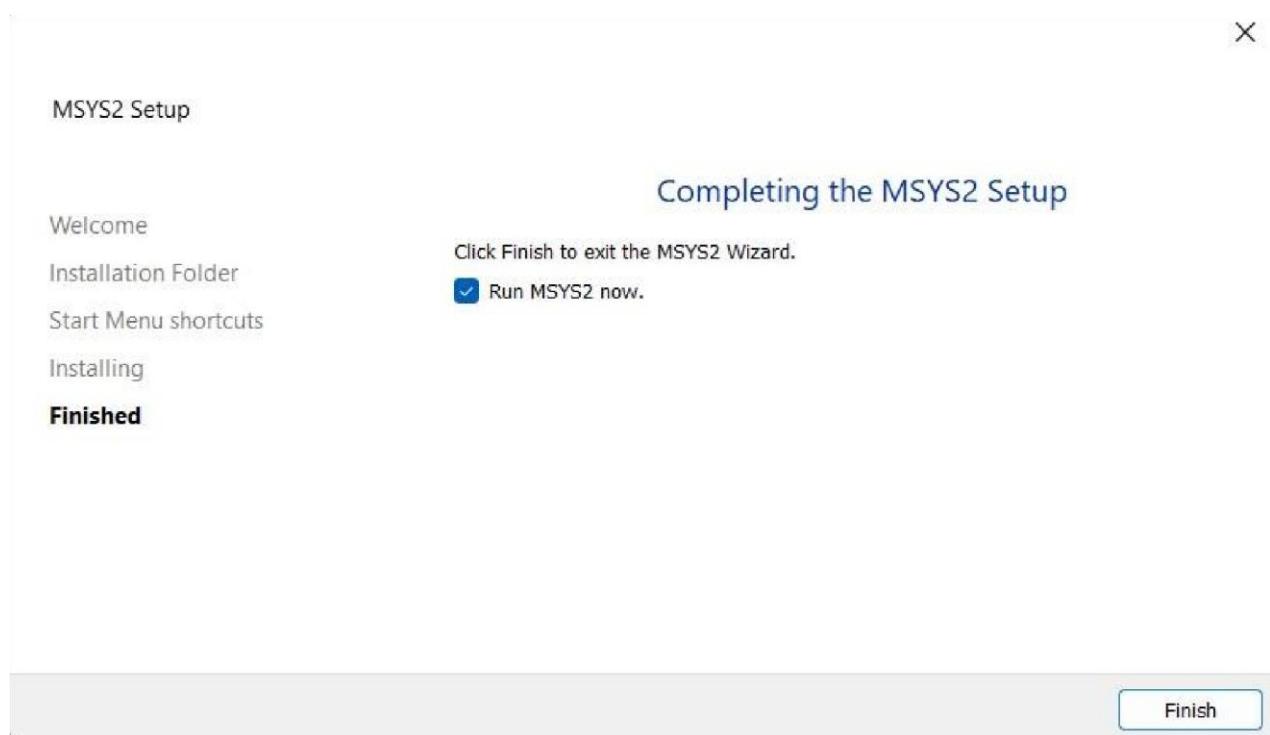
```
PS C:\Users\ARADHYA> g++ --version
g++ : The term 'g++' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the
spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ g++ --version
+ ~~~
+ CategoryInfo          : ObjectNotFound: (g++:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
```

3. Download *msys2-x86_6420240113.exe* from

https://github.com/msys2/msys2-installer/releases/download/2024-01-13/msys2-x86_64-20240113.exe and execute it to install msys2-x86 MinGW compiler.

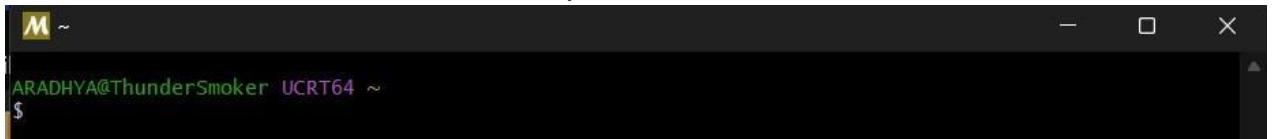
Note: Choose all default options while installing

You should get the below output



Check the “Run MSYS2 now” checkbox and finish.

A MSYS2 UCRT64 terminal should open which confirms the installation.



```
M ~
ARADHYA@ThunderSmoker UCRT64 ~
$
```

4. In this terminal, install the MinGW-w64 toolchain by running the following command:

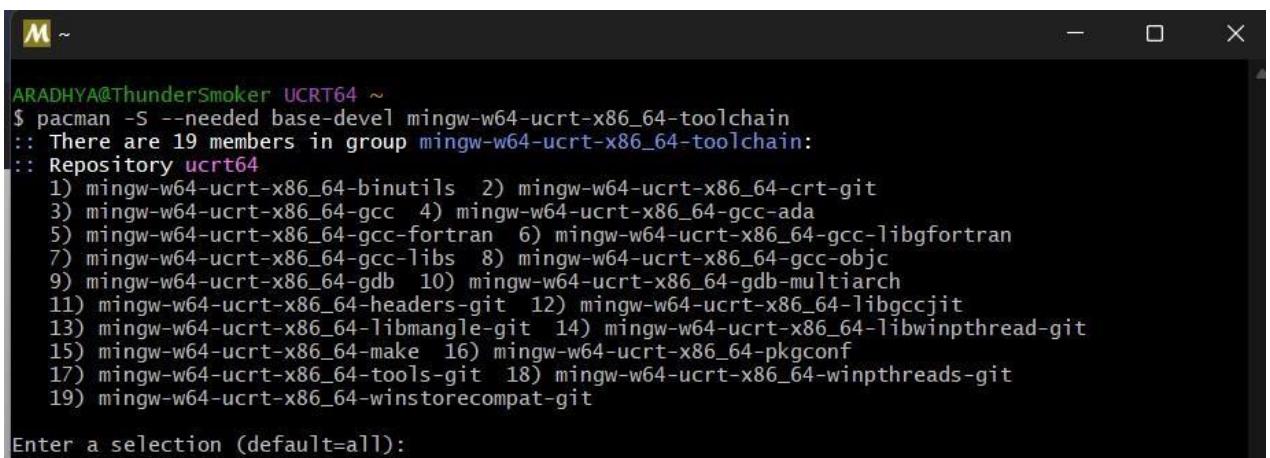
```
pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
```

pressing **Enter**

Enter when prompted whether to proceed with the installation. **Y**

5. Accept the default number of packages in the **toolchain** group by

6.



```
M ~
ARADHYA@ThunderSmoker UCRT64 ~
$ pacman -S --needed base-devel mingw-w64-ucrt-x86_64-toolchain
:: There are 19 members in group mingw-w64-ucrt-x86_64-toolchain:
:: Repository ucrt64
 1) mingw-w64-ucrt-x86_64-binutils  2) mingw-w64-ucrt-x86_64-crt-git
 3) mingw-w64-ucrt-x86_64-gcc  4) mingw-w64-ucrt-x86_64-gcc-ada
 5) mingw-w64-ucrt-x86_64-gcc-fortran  6) mingw-w64-ucrt-x86_64-gcc-libfortran
 7) mingw-w64-ucrt-x86_64-gcc-libs  8) mingw-w64-ucrt-x86_64-gcc-objc
 9) mingw-w64-ucrt-x86_64-gdb  10) mingw-w64-ucrt-x86_64-gdb-multiarch
 11) mingw-w64-ucrt-x86_64-headers-git  12) mingw-w64-ucrt-x86_64-libgccjit
 13) mingw-w64-ucrt-x86_64-libmangle-git  14) mingw-w64-ucrt-x86_64-libwinpthread-git
 15) mingw-w64-ucrt-x86_64-make  16) mingw-w64-ucrt-x86_64-pkgconf
 17) mingw-w64-ucrt-x86_64-tools-git  18) mingw-w64-ucrt-x86_64-winthreads-git
 19) mingw-w64-ucrt-x86_64-winstorecompat-git

Enter a selection (default=all):
```

7. Add the path to your MinGW-w64 **bin** folder to the

Windows **PATH** environment variable by using the following steps:

- In the Windows search bar, type **Settings** to open your Windows Settings.
- Search for **Edit environment variables for your account**.

- c. In your **User variables**, select the **Path** variable and then select **Edit**.
- d. Select **New** and add the MinGW-w64 destination folder you recorded during the installation process to the list. If you used the default settings above, then this will be the path: **C:\msys64\ucrt64\bin**.
- e. Select **OK** to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available.

Check your MinGW installation

To check that your MinGW-w64 tools are correctly installed and available, open a **new** Command Prompt and type:

```
gcc --version  
g++ --version  
gdb --version
```

You should see output that states which versions of GCC, g++ and GDB you have installed. If this is not the case:

1. Make sure your PATH variable entry matches the MinGW-w64 binary location where the toolchain was installed. If the compilers do not exist at that PATH entry, make sure you followed the previous instructions.
2. If **gcc** has the correct output but not **gdb**, then you need to install the packages you are missing from the MinGW-w64 toolset. If on compilation you are getting the "The value of miDebuggerPath is

invalid." message, one cause can be you are missing the **mingww64-gdb** package.

Great !! Now you have installed C++ compiler in your system.

- Now, lets head on to OpenCV installation.

Copy the path of **path-to-build\x64\mingw\bin** folder path from the extracted zip provided and add to Windows PATH environment variable.

- Now let's compile our solution.cpp present inside the extracted zip folder with the below command in terminal:

```
g++ -o solution solution.cpp I path-to-build\include -L path-to-build\x64\mingw\bin  
-lopencv_core490 -lopencv_highgui490 -lopencv_imgcodecs490 -lopencv_stitching490 -lopencv_features2d490 -lopencv_imgproc490 -lopencv_xfeatures2d490 -lopencv_calib3d490
```

Example Usage:-

```
PS C:\Users\ARADHYA\OneDrive\Desktop\BOSC\Bosch Auto Vision> g++ -o solution solution.cpp -I "C:\Users\ARADHYA\OneDrive\Desktop\BOSC\Bosch Auto Vision\Bosch Auto Vision\build\include" -L "C:\Users\ARADHYA\OneDrive\Desktop\BOSC\Bosch Auto Vision\Bosch Auto Vision\build\x64\mingw\bin" -lopencv_core490 -lopencv_highgui490 -lopencv_imgcodecs490 -lopencv_stitching490 -lopencv_features2d490 -lopencv_imgproc490 -lopencv_xfeatures2d490 -lopencv_calib3d490
```

- Last Step , Executing the compiled solution.exe to produce a stitched panoramic image.

Run any of below commands according to need through terminal in the folder which contains previously compiled solution.exe and all four images.

- a.** This command takes an average of 5.86 sec with comparatively poor blending quality.

```
./solution.exe img1.jpg img2.jpg img3.jpg img4.jpg --features sift --matcher homography --estimator homography --match_conf 0.4 --wave_correct horiz --seam gc_color --blend multiband --blend_strength 20
```

- b.** This command takes an average of 22.70 sec with comparatively better blending quality.

```
./solution.exe img1.jpg img2.jpg img3.jpg img4.jpg --features sift --matcher homography --estimator homography --match_conf 0.6 --wave_correct horiz --seam gc_color --blend multiband --blend_strength 20 --work_megapix 6 --seam_megapix 0.5
```

After execution is completed you can view the result image produced in the same directory with the name **result.jpg**.

The parameters chosen in above commands, are chosen best based on trial and error method.

You can see the parameter usage guide by running:

```
./solution.exe
```

Note:- This result.jpg is the final stitched panoramic image of all four images provided. You can change the number of required images and change the execution command accordingly.

Approach and Concepts used in creating this solution can be read in Solution&Approach.pdf.