0 参考博客：

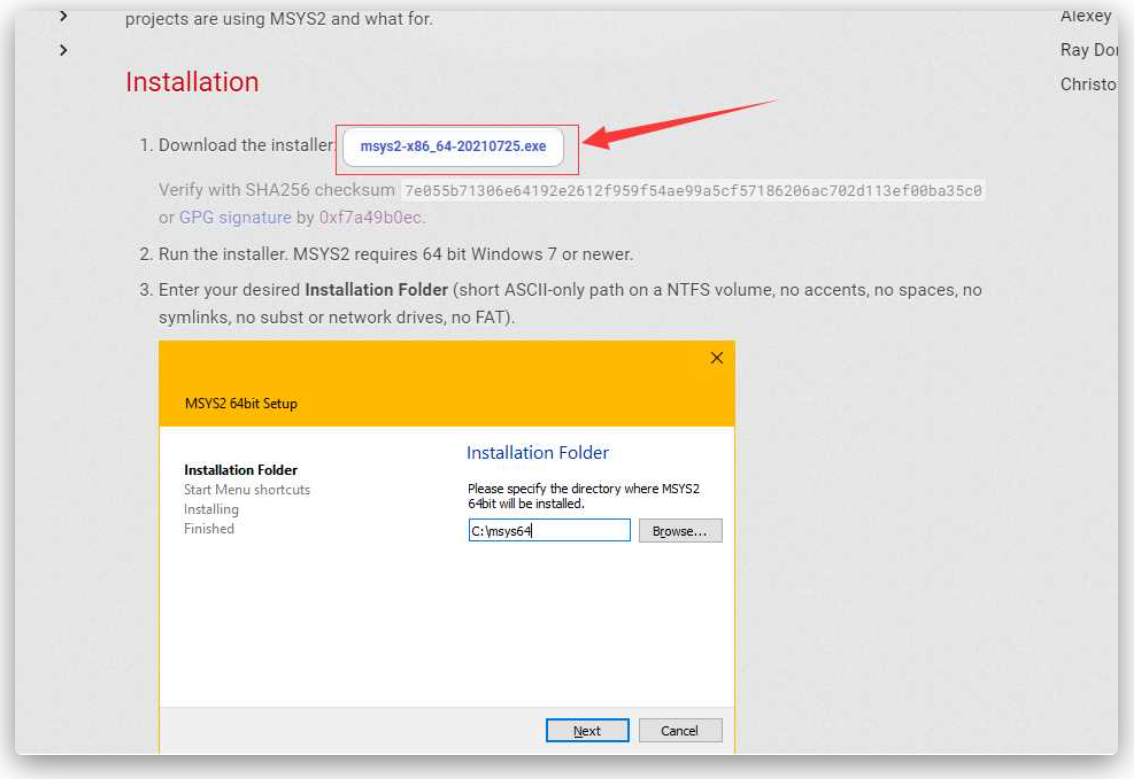
https://www.stubbornhuang.com/1555/

2 编译步骤

2.1 MSYS2环境配置

MSYS2官网：https://www.msys2.org/

2.1.1 点击以下按钮下载msys2



点击安装，自行设置安装位置

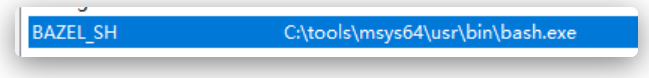
2.1.2 配置系统环境变量

如安装位置为：D:\Program Files (x86)\msys64，则将D:\Program Files (x86)\msys64\usr\bin添加到系统变量中

图形用户界面, 文本, 应用程序

描述已自动生成

根据网友大佬的反馈，如果使用的系统为Windows11，则还需要明确在系统环境变量中指定BAZEL\_SH为：D:\Program Files (x86)\msys64\usr\bin\bash.exe，示例设置图为：

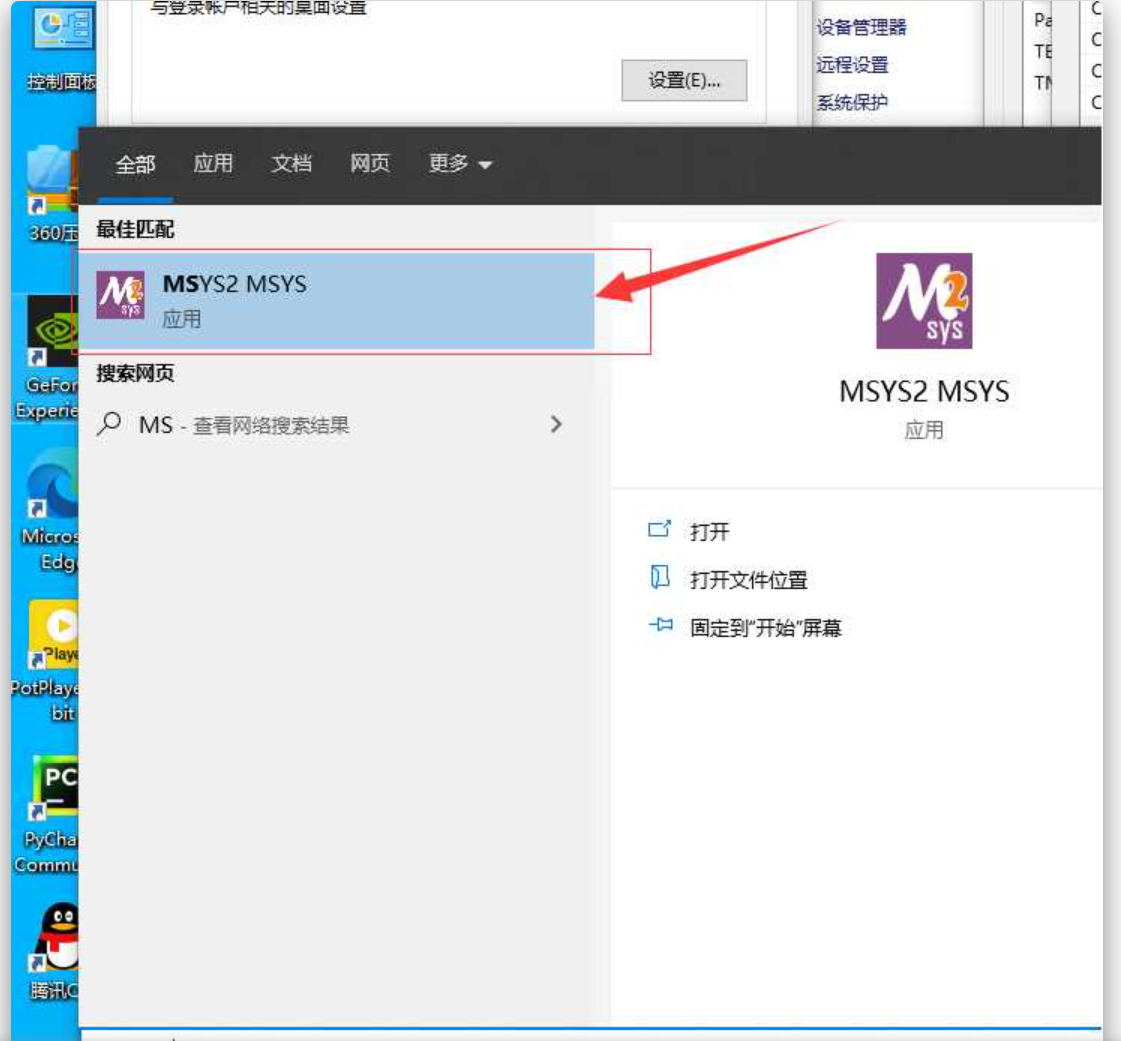


不然编译时会出现错误，具体可参考：

* [https://github.com/google/mediapipe/issues/3459#issuecomment-1167468953](https://github.com/google/mediapipe/issues/3459" \l "issuecomment-1167468953" \t "_blank" \o "https://github.com/google/mediapipe/issues/3459#issuecomment-1167468953)
* [https://docs.bazel.build/versions/main/install-windows.html#bazel-does-not-find-bash-or-bashexe](https://docs.bazel.build/versions/main/install-windows.html" \l "bazel-does-not-find-bash-or-bashexe" \t "_blank" \o "https://docs.bazel.build/versions/main/install-windows.html#bazel-does-not-find-bash-or-bashexe)

2.1.3 升级msys2包数据库

打开MSYS2 MSYS



在打开的MSYS命令行中输入：pacman -Syu

图形用户界面

低可信度描述已自动生成

等待各种包下载升级安装

2.1.4 升级msys2基础数据库

继续在MSYS中输入：pacman -Su

图形用户界面

低可信度描述已自动生成

等待升级完成

2.1.5 下载安装必要的包：unzip

MSYS2没有内置unzip，需要手动安装。

打开cmd.exe，输入以下命令：pacman -S git patch unzip

图片包含 图形用户界面

描述已自动生成

等待下载安装完成即可，经过以上步骤，MSYS2的配置就完成了

2.2 Python环境配置

需要在电脑上安装python，并且配置系统环境，具体步骤就不再赘述。在Python环境中需要装好tensorflow cpu，如果准备编译GPU版本需要安装gpu版本的tensorflow。

如果在电脑上已经使用Anaconda创建虚拟环境，可以使用Anaconda的base环境或者其中的虚拟环境，我是用的就是Anaconda的base环境。

2.3 安装VS2019和WinSDK

下载免费的社区版即可，在安装vs2019的过程中会自动安装WinSDK

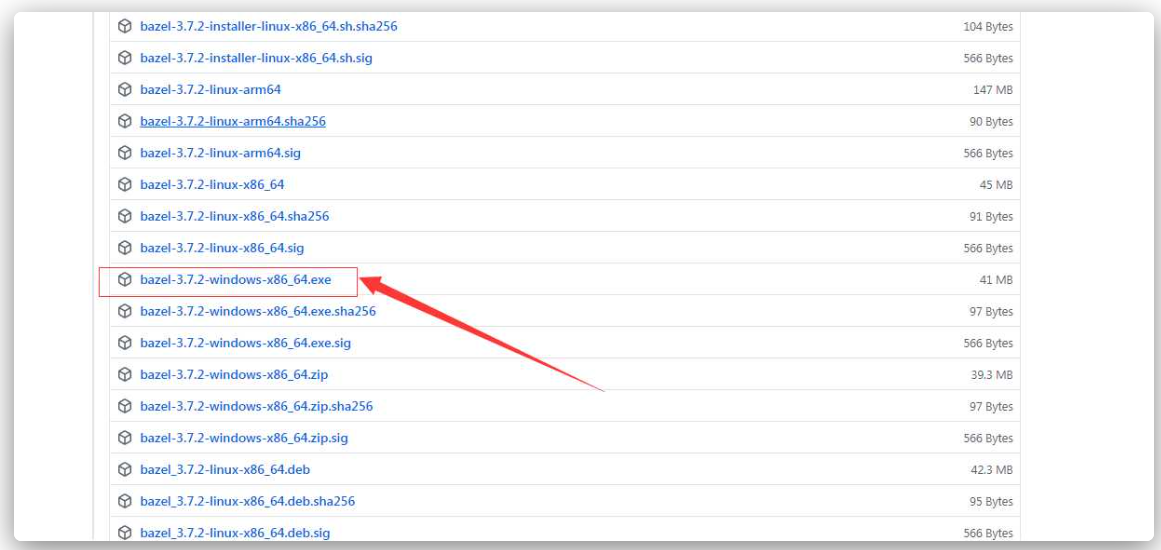
2.4 配置Bazel

2.4.1 Bazel下载

官方文档的建议是下载Bazel 3.7.2或者更高版本，这里选择Bazel 3.7.2版本下载。

Bazel下载地址：https://github.com/bazelbuild/bazel/releases

在该页面使劲往下翻，翻到3.7.2下载bazel-3.7.2-windows-x86\_64.exe



2.4.2 Bazel系统环境配置

下载的文件名为：bazel-3.7.2-windows-x86\_64.exe，为了之后可以更好地在cmd.exe中使用，可以拷贝一个副本，并将其改名为bazel.exe，这样在命令行调用的时候只需要输入bazel而不需要输入bazel-3.7.2-windows-x86\_64。

如果bazel.exe保存的路径为：F:\软件安装包\bazel，需要将该路径添加到系统环境变量中，使其可以在命令行中调用，需要将其路径添加到系统环境变量

图形用户界面, 文本, 应用程序

描述已自动生成

系统环境配置完成之后，在cmd.exe输入bazel，如出现以下提示则说明配置成功。

文本

描述已自动生成

2.5 设置Bazel变量

根据官方文档，需要设置Bazel的编译变量：

# Please find the exact paths and version numbers from your local version.

C:\> set BAZEL\_VS=C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools

C:\> set BAZEL\_VC=C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC

C:\> set BAZEL\_VC\_FULL\_VERSION=<Your local VC version>

C:\> set BAZEL\_WINSDK\_FULL\_VERSION=<Your local WinSDK version>

文本

描述已自动生成

其中：

* BAZEL\_VS ： vs2019的安装目录，比如我的：D:\Program Files (x86)\Microsoft Visual Studio\2019\Community
* BAZEL\_VC：vs2019安装目录下C++主要目录，比如我的 BAZEL\_VC=D:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC
* BAZEL\_VC\_FULL\_VERSION ： VS2019的中vc的版本号，

该版本号可以在VS2019的安装目录D:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Redist\MSVC下找到：

图形用户界面, 应用程序

描述已自动生成

图片中的14.29.30036就是VC的版本号。

BAZEL\_WINSDK\_FULL\_VERSION：WinSDK的版本号，这可以在控制面板-程序-程序和功能中查看，找到Windows Software Development Kit，后面跟的数字就是WinSDK的版本号，如下图所示

图形用户界面, 应用程序

描述已自动生成

那么最后我的配置项为：

set BAZEL\_VS=D:\Microsoft\Microsoft Visual Studio\2019\Community

set BAZEL\_VC=D:\Microsoft\Microsoft Visual Studio\2019\Community\VC

set BAZEL\_VC\_FULL\_VERSION=14.29.30133

set BAZEL\_WINSDK\_FULL\_VERSION=10.0.19041.685

图形用户界面, 文本

中度可信度描述已自动生成

需要在MSYS上设置Bazel变量而不是在cmd.exe中，在MSYS中依次输入上述命令，这里有个坑，Program Files (x86)的这个括号需要进行转义，所以应该变成Program Files (x86)

文本

描述已自动生成

2.6 下载opencv

官方文档指出目前使用OpenCV 3.4.10，所以下载OpenCV 3.4.10即可。

下载地址：<https://opencv.org/releases/>

图形用户界面

描述已自动生成

下载完成之后，点击exe，解压到自己想要的目录，这里我的解压目录是：F:\opencv-3.4.10-vc14\_vc15，不要使用中文路径，不然在后面编译会出错。

**2.7 Mediapipe仓库克隆以及bazel的编译文件WORKSPACE的修改**

使用git迁移Mediapipe官方仓库：

git clone https://github.com/google/mediapipe.git

Bash

Copy

迁移完成之后，打开该仓库根目录下的WORKSPACE文件，并在该文件中搜索“windows\_opencv”，并将OpenCV的路径修改为你本地保存OpenCV的路径

修改后：

new\_local\_repository(

name = "windows\_opencv",

build\_file = "@//third\_party:opencv\_windows.BUILD",

path = "F:\\opencv-3.4.10-vc14\_vc15\\opencv\\build",

)

Bash

Copy

保存即可。

**2.8 编译并运行hello world**

官方文档提示，如果需要在Windows上编译Mediapipe需要添加

--action\_env PYTHON\_BIN\_PATH="C://path//to//python.exe"

Bash

Copy

在编译命令行中，这里的PYTHON\_BIN\_PATH就是上述步骤中所安装的python路径，配置好即可。

**2.8.1 编译**

使用cmd.exe将目录切换到Mediapipe仓库的根目录下，然后运行以下命令：

bazel build -c opt --define MEDIAPIPE\_DISABLE\_GPU=1 --action\_env PYTHON\_BIN\_PATH="D:\\Anaconda\\python.exe" mediapipe/examples/desktop/hello\_world

Bash

Copy

上述的PYTHON\_BIN\_PATH需要修改为你自己的python路径，编译过程中需要从github上拉取一些库，所以最好科学上网，不然会出现can't fetch错误。

编译完成可以看到以下信息：

文本

描述已自动生成

**2.8.2 运行**

编译完成之后，继续在cmd.exe中依次输入以下命令：

set GLOG\_logtostderr=1

bazel-bin\mediapipe\examples\desktop\hello\_world\hello\_world.exe

Bash

Copy

如果编译成功，运行结果如下

文本

描述已自动生成

如果能够编译并且运行成功hello world示例程序，那么说明你的Mediapipe的Windows C++编译环境配置完成，之后的就尽情的在上面做魔改吧！！！