

## 亮牛SDK交叉编译环境说明

### 软件列表

#### 软件安装 (Windows10 x64示例)

Python3

ARM GCC编译器套件

CMake

Ninja

GNU MCU Eclipse Windows Build Tools

SEGGER JLink 工具

Visual Studio Code 编辑器

#### 软件安装 (Ubuntu 20.04 x64 示例)

使用包管理器安装必要软件

官网下载其他必要软件 (Ubuntu 版本)

ARM GCC 编译器套件

SEGGER JLink 工具

Visual Studio Code 编辑器

# 亮牛SDK交叉编译环境说明

---

- 亮牛 SDK 支持 **CMake** + **ARM GCC** 编译;
- 搭配 SEGGER **JLink**, 可实现烧录, 可启动GDB调试服务器;
- 使用 Visual Studio Code 编辑器加载 **Cortex-Debug** 插件可在线调试;

这里以 windows10 x64 和 Ubuntu 20.04 x64 进行说明。

## 软件列表

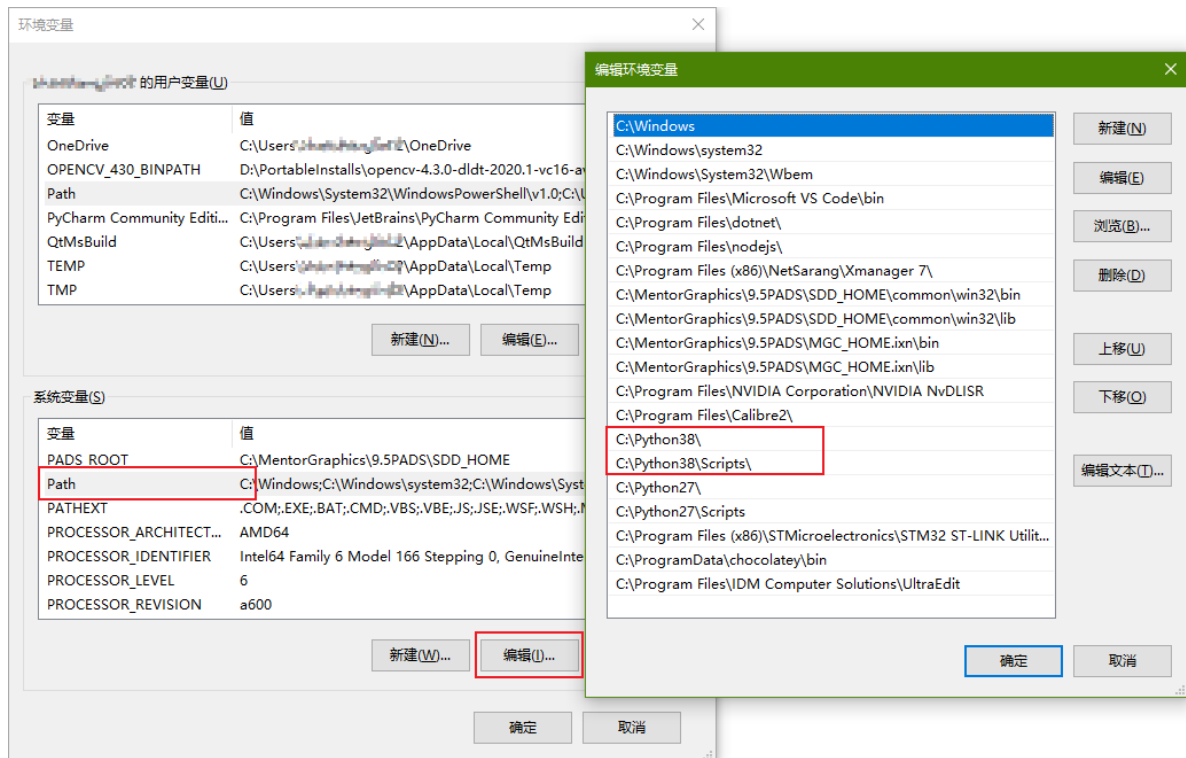
---

软件	简介
Python3 (3.6 及以上版本)	构建中使用 python 脚本，且SDK路径 <b>不能</b> 包含中文，Python2.x 不支持。
ARM GCC 编译器套件	选用ARM官方的 <b>GNU Arm Embedded Toolchain: 10-2020-q4-major</b> 版本。
CMake	根据所选生成器生成对应的 Makefile 文件或者 build.ninja 文件。
Ninja (推荐)	类似于 Make 的构建工具，处理 CMake 生成的 <b>build.ninja</b> 文件，速度比 <b>Make</b> 快很多。
Make (Linux 版本)	读取 CMake 生成的 Makefile 文件，调用编译器套件生成目标。
GNU MCU Eclipse Windows Build Tools (Windows 版本)	它是 GNU Make的 windows 版本，调用GCC编译套件执行真正的编译动作，生成可执行镜像文件。
SEGGER JLink	它可以烧录固件，也可以启动GDB调试服务器。 ( <b>注意</b> 推荐安装 <b>V752d</b> 及以下版本)
Visual Studio Code 编辑器	可选，但以gdb调试时，必须安装此编辑器以及附加插件。

## 软件安装（Windows10 x64示例）

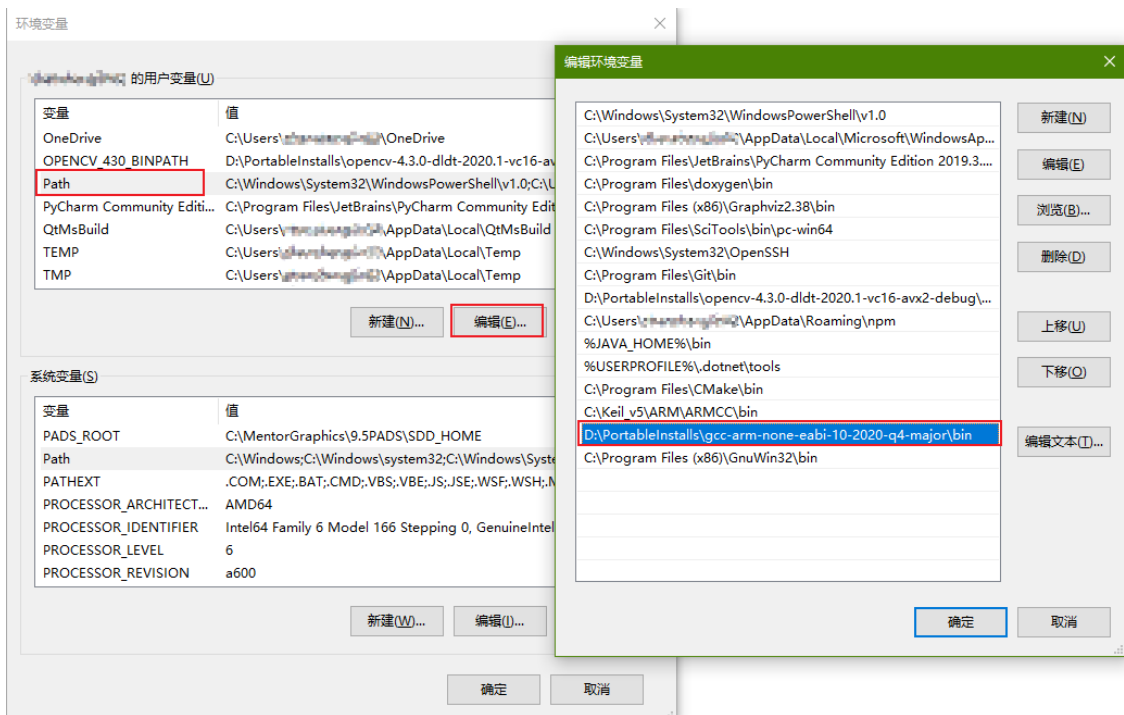
### Python3

1. 安装：双击Python3安装包，以默认方式安装；
2. **PATH** 环境变量：打开环境变量编辑器，编辑用户或者系统的环境变量**PATH**，添加对应的 python3 安装路径（可放在 Python2 路径后面），示例如下：
3. 进入到python3安装路径，复制 `python.exe` 并重新命名为 `python3.exe`，在新打开的命令行中输入 `python3 --version` 检查确认版本信息；

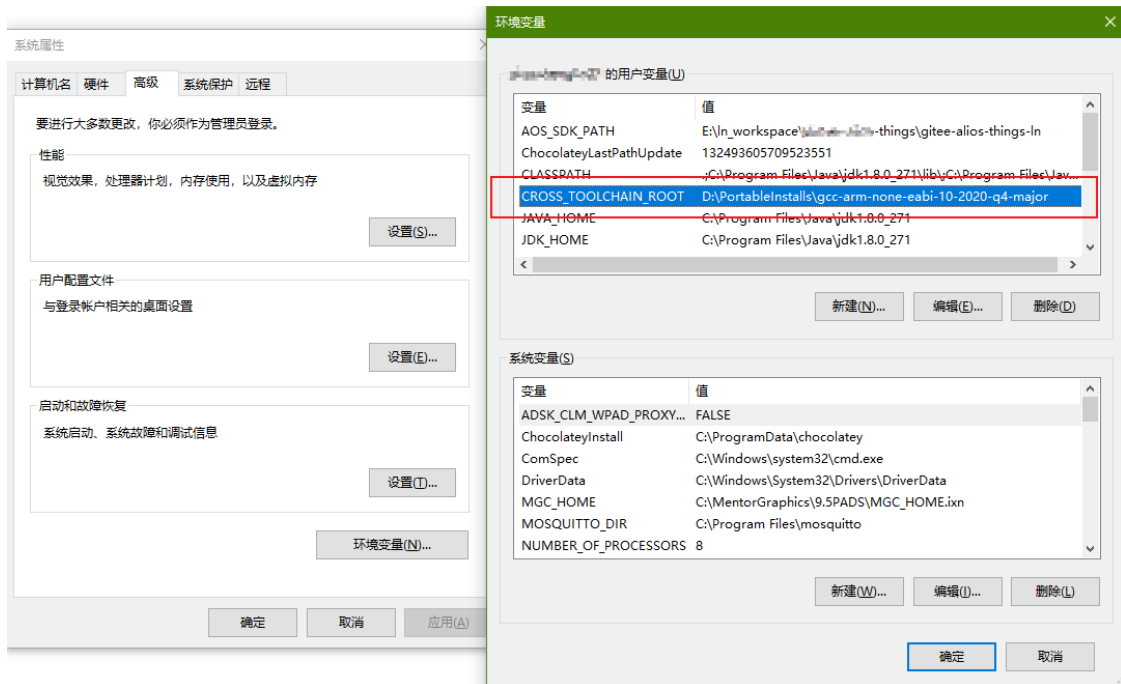


## ARM GCC编译器套件

1. 解压: **gcc-arm-none-eabi-10-2020-q4-major-win32.zip** 是绿色版本, 无需安装, 解压到某个目录即可, 例如解压到 **D:\PortableInstalls** 目录;
2. **PATH** 环境变量: 示例如下

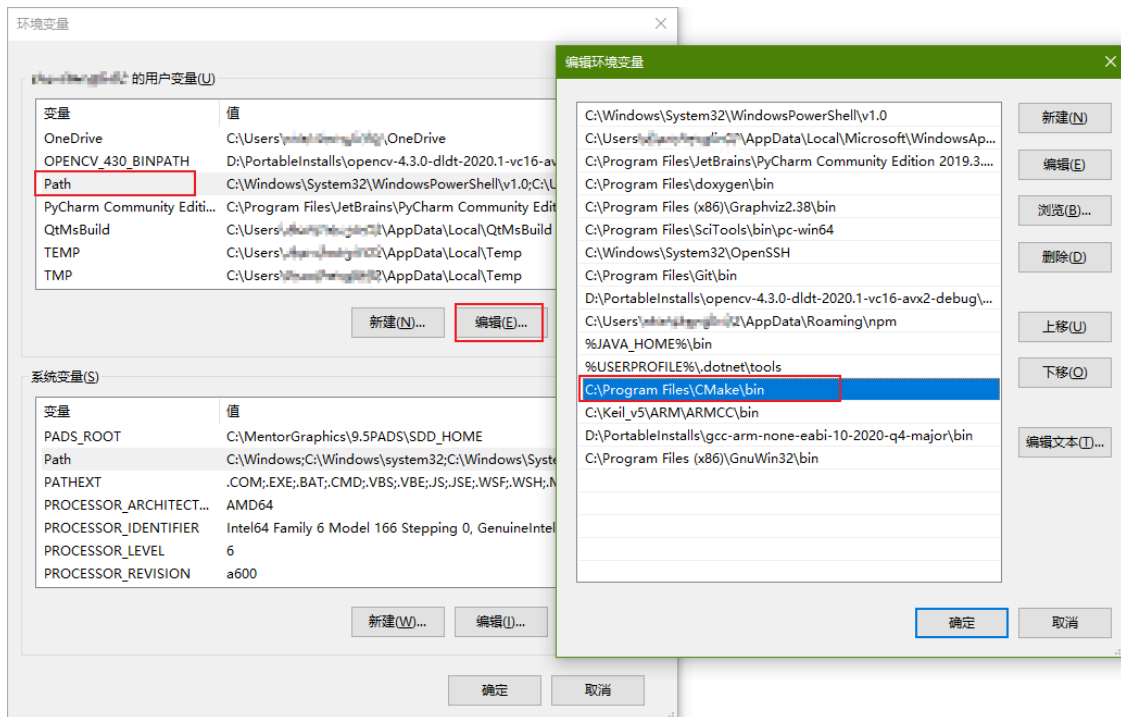


3. 检查: 在新打开的命令行中输入 **arm-none-eabi-gcc --version** 确认版本信息;
4. 再添加一个环境变量 **CROSS\_TOOLCHAIN\_ROOT**, 它的值为 **D:\PortableInstalls\gcc-arm-none-eabi-10-2020-q4-major**, 如下所示:



## CMake

1. 安装：双击安装包，以默认方式安装；
2. PATH 环境变量：示例如下

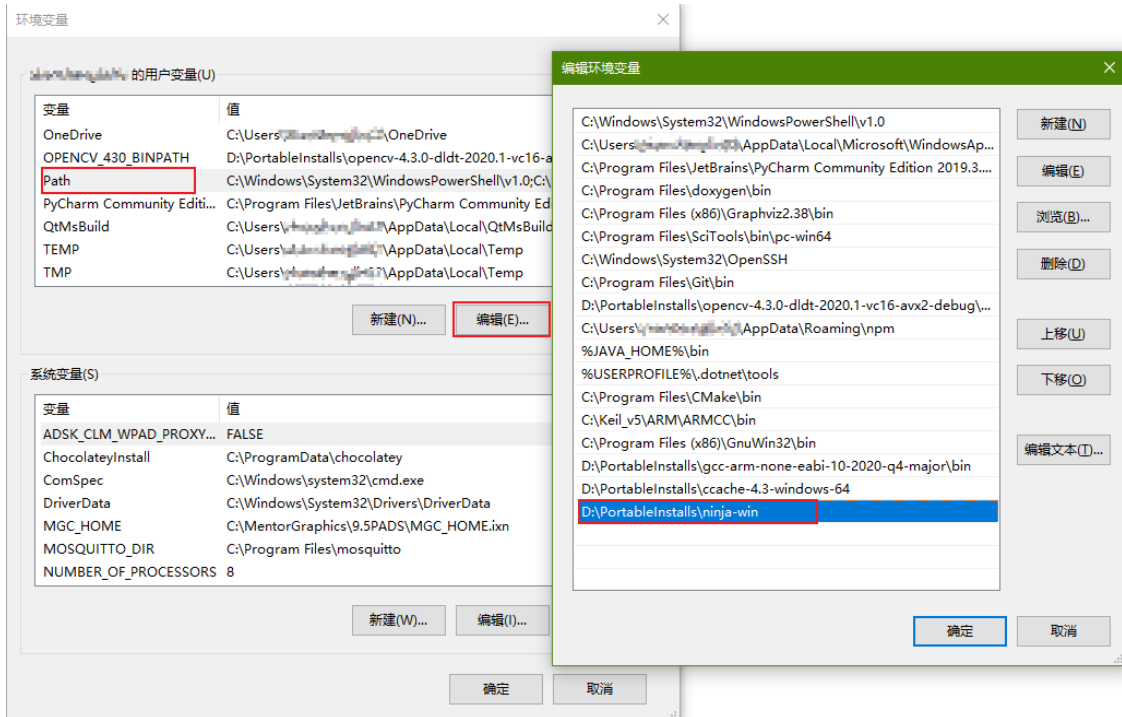


3. 检查：在新打开的命令行中输入 `cmake --version` 确认版本信息；

## Ninja

1. 安装：绿色版本，解压到某个目录即可；

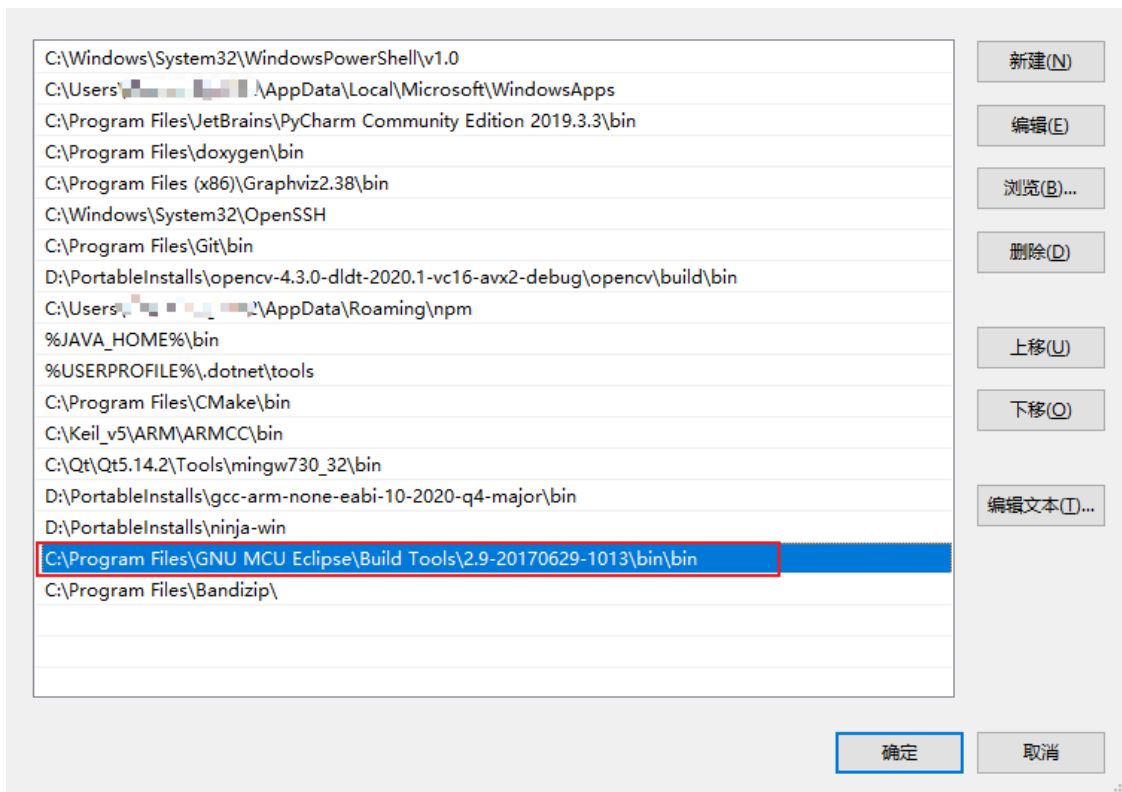
## 2. PATH 环境变量：示例如下：



## 3. 检查：在新打开的命令行中输入 `ninja --version` 确认版本信息；

# GNU MCU Eclipse Windows Build Tools

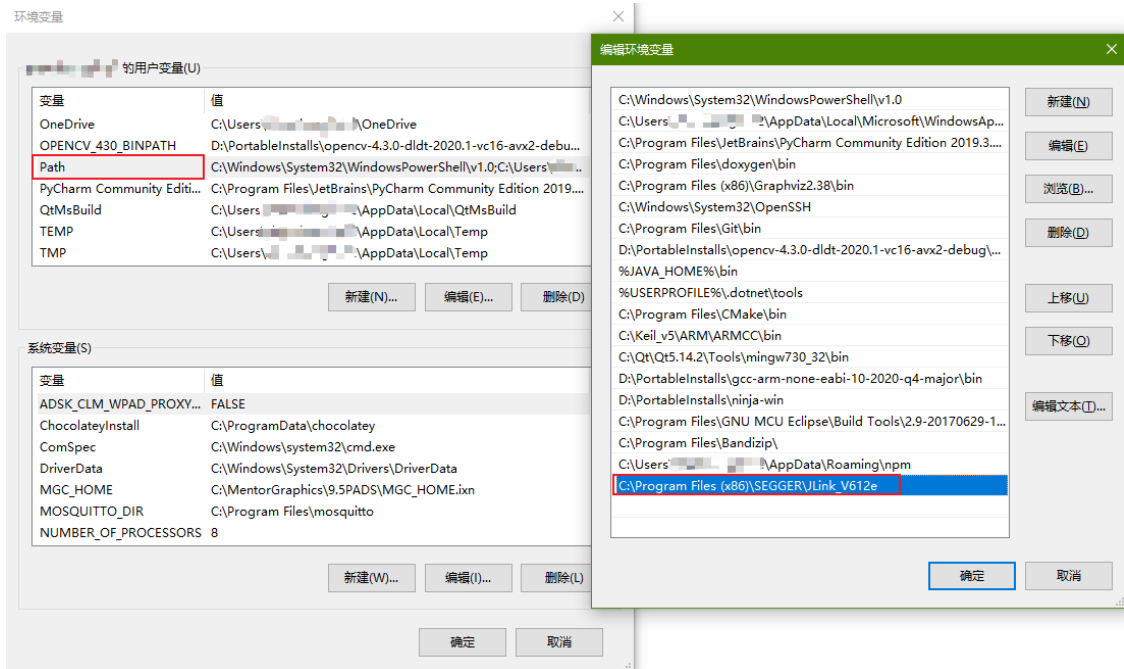
1. 安装：双击安装；
2. PATH 环境变量：如下：



## 3. 检查：在新打开的命令行中输入 `make --version` 确认版本信息；

# SEGGER JLink 工具

1. 安装：双击安装，默认选项安装；
2. PATH 环境变量：如下所示：



3. 检查：在新打开的命令行输入 `JFlash.exe -?`，如果能弹出窗口即可。

## Visual Studio Code 编辑器

Visual Studio Code 编辑器是一款轻量级、功能强大的源码编辑器，配合各种插件可以实现类似IDE的效果。

推荐安装以下插件以便编辑源码、调试嵌入式程序：

1. **C/C++ IntelliSense**
2. **CMake**
3. **CMake Tools**
4. **Cortex-Debug** 它的配置文件参考[cortex-debug](#)

## 软件安装（Ubuntu 20.04 x64 示例）

### 使用包管理器安装必要软件

```
sudo apt-get install python3 cmake ninja-build make
```

# 官网下载其他必要软件 (Ubuntu 版本)

## ARM GCC 编译器套件

从下载页面选择 [gcc-arm-none-eabi-10-2020-q4-major-x86\\_64-linux.tar.bz2](https://developer.arm.com/architectures/datasheets/gcc-linaro-versions/10-2020-q4-major/x86_64-linux.tar.bz2)

解压到某个目录，然后在 `~/.bashrc` 文件末尾导出环境变量

`CROSS_TOOLCHAIN_ROOT`：

```
1 # GCC ARM NONE EABI
2 export CROSS_TOOLCHAIN_ROOT=$HOME/PortableInstalls/gcc-arm-
  none-eabi-10-2020-q4-major
```

新打开一个命令行或者在当前命令行输入 `source ~/.bashrc` 即可重新加载环境变量。

## SEGGER JLink 工具

从下载页面选择 [v7.52d 64-bit DEB Installer](#)

输入命令安装

```
sudo dpkg -i JLink_Linux_V752d_x86_64.deb
```

## Visual Studio Code 编辑器

从下载页面选择 [.deb 64bit](#)

输入命令安装

```
sudo dpkg -i code_1.59.1-1629375198_amd64.deb
```

打开 vscode 并安装如下插件：

1. **C/C++ IntelliSense**
2. **CMake**
3. **CMake Tools**
4. **Cortex-Debug** 它的配置文件参考[cortex-debug](#)