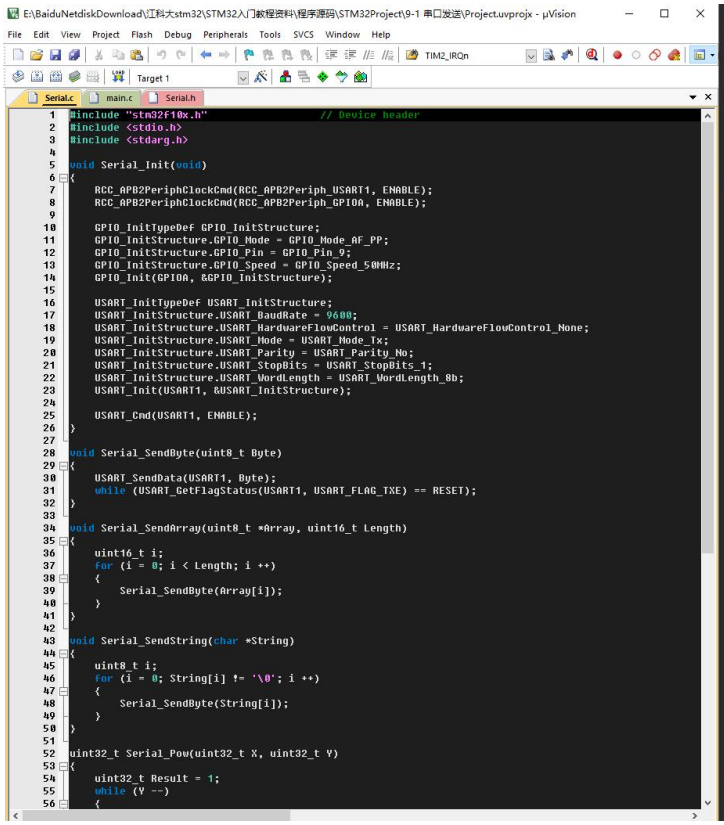
VSCODE 和 KEIL 协同使用开发 stm32 程序

VSCODE 是一款广受好评的代码编辑器，KEIL 是常用的嵌入式开发工具但编程界面简陋。将两个工具一起搭配使用，能大大提高我们的效率。

你可以把 VSCODE 专门用来编辑和编译，KEIL 用来对文件进行增删下载配置环境。原始的 KEIL 代码编辑界面在编辑和阅读上都十分的不方便，只要你用过 VSCODE 的编辑界面就再也不想回去使用 KEIL 的代码编辑界面了。如下是同一份代码在 KEIL 和 VSCODE 上呈现的不同效果。KEIL 的阅读和编辑体验是远远不如 VSCODE。

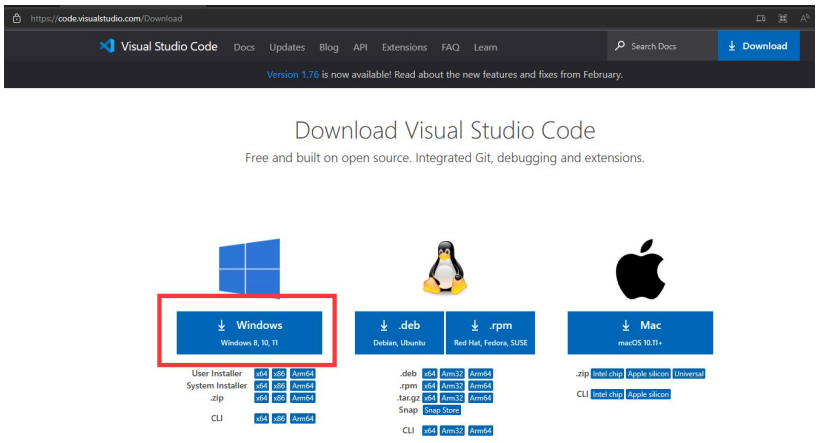


```c
#include "stm32f10x.h"                  // Device header
#include <stdio.h>
#include <stdarg.h>

void Serial_Init(void)
{
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

    GPIO_InitTypeDef GPIO_InitStructure;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    USART_InitTypeDef USART_InitStructure;
    USART_InitStructure.USART_BaudRate = 9600;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Tx;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_Init(USART1, &USART_InitStructure);

    USART_Cmd(USART1, ENABLE);
}

void Serial_SendByte(uint8_t Byte)
{
    USART_SendData(USART1, Byte);
    while (USART_GetFlagStatus(USART1, USART_FLAG_TXE) == RESET);
}

void Serial_SendArray(uint8_t *Array, uint16_t Length)
{
    uint16_t i;
    for (i = 0; i < Length; i ++)
    {
        Serial_SendByte(Array[i]);
    }
}

void Serial_SendString(char *String)
{
    uint8_t i;
    for (i = 0; String[i] != '\0'; i ++)
    {
        Serial_SendByte(String[i]);
    }
}

uint32_t Serial_Pow(uint32_t X, uint32_t Y)
{
    uint32_t Result = 1;
    while (Y --)
    {
```

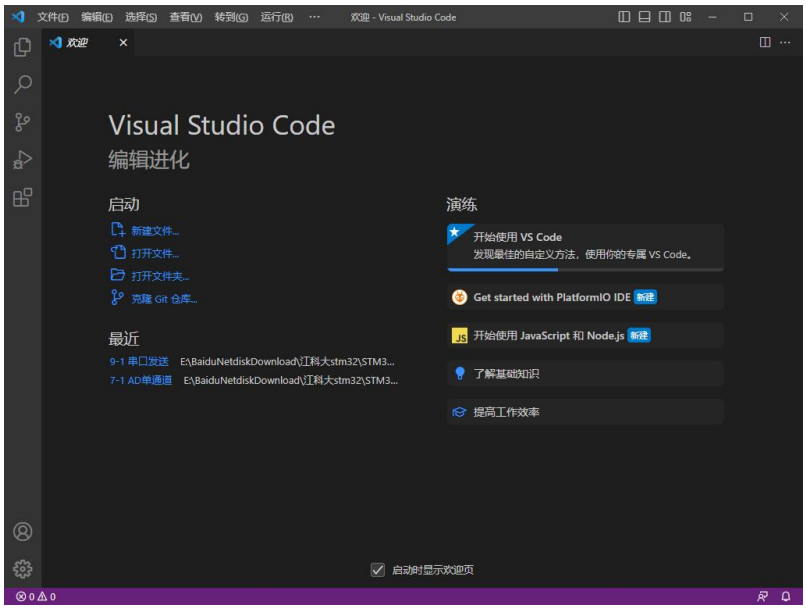
1、下载 VSCODE

可以从其官网上下载，地址是

Download Visual Studio Code - Mac, Linux, Windows



下载完成后，界面如下所示，你可以把 VSCODE 设想成专门用来编辑代码文件的 WORD 编辑器，如果你会用 WORD 写文档，那么你也就也能一定会学会用 VSCODE 编辑代码。不过比 WORD 更复杂一点的是 VSCODE 可以安装各种各样的扩展来专门针对特定的编程语言。



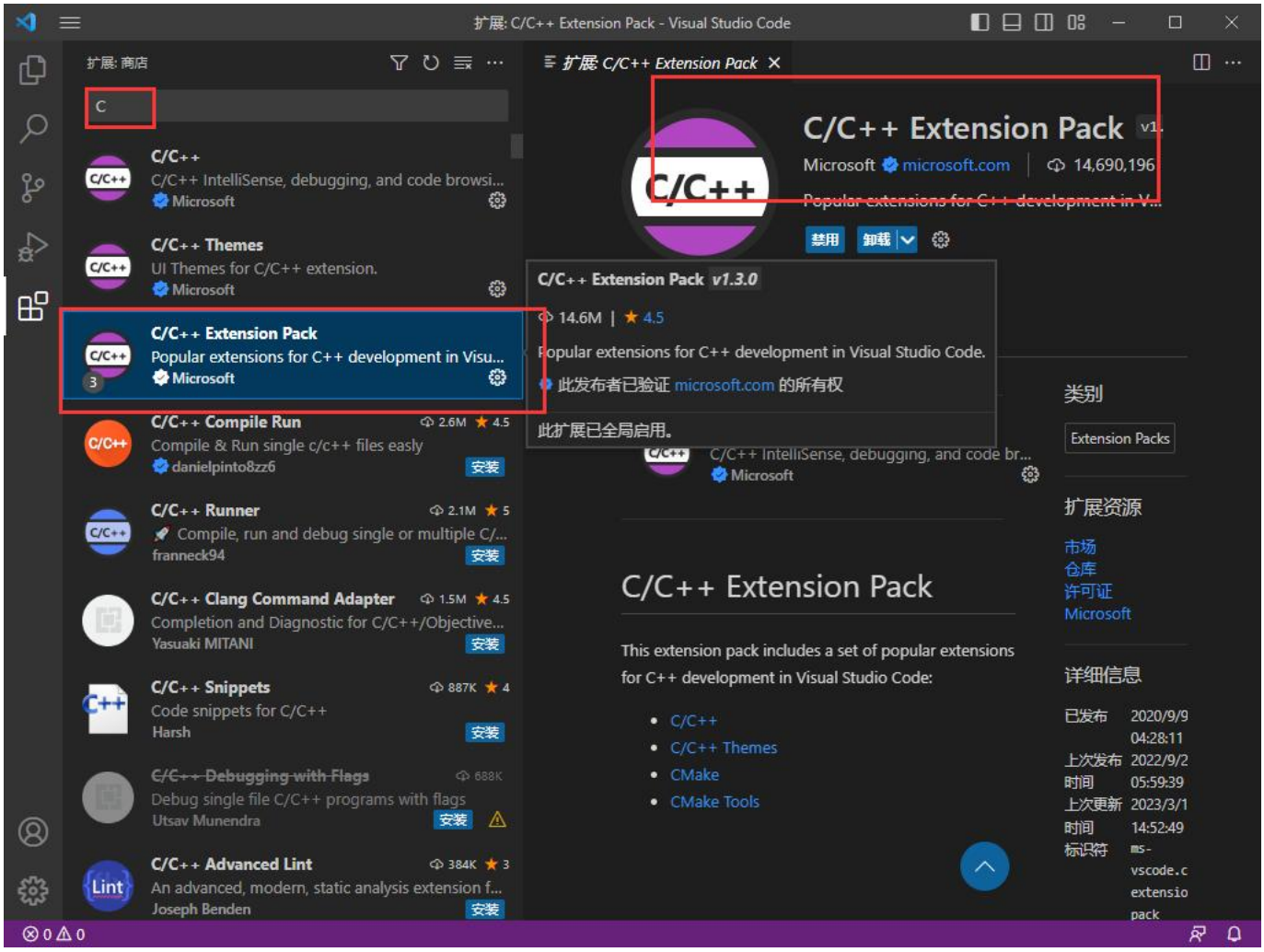
2、安装扩展

2.1 安装汉语扩展

安装 VSCODE 打开后默认语言是英语，你可以在扩展安装选项下载一个汉语扩展让 VSCODE 界面变成中文。选中中文扩展安装后再重启一下 VSCODE,你的界面就变成中文的了。

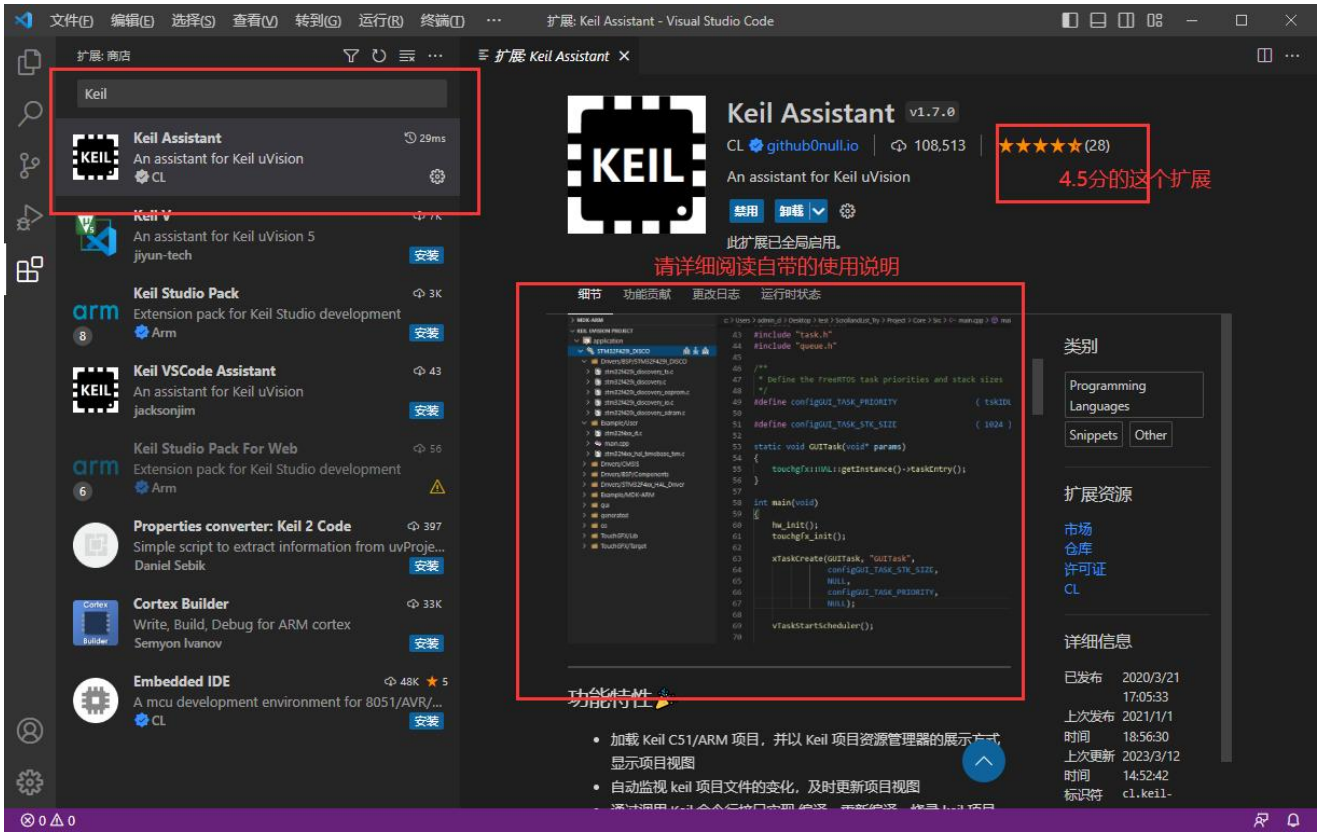
2.2 安装 C/C++的扩展

同样的，在扩展商店的输入框上写个 C,找到 C/C++ Extension Pack,安装上它。

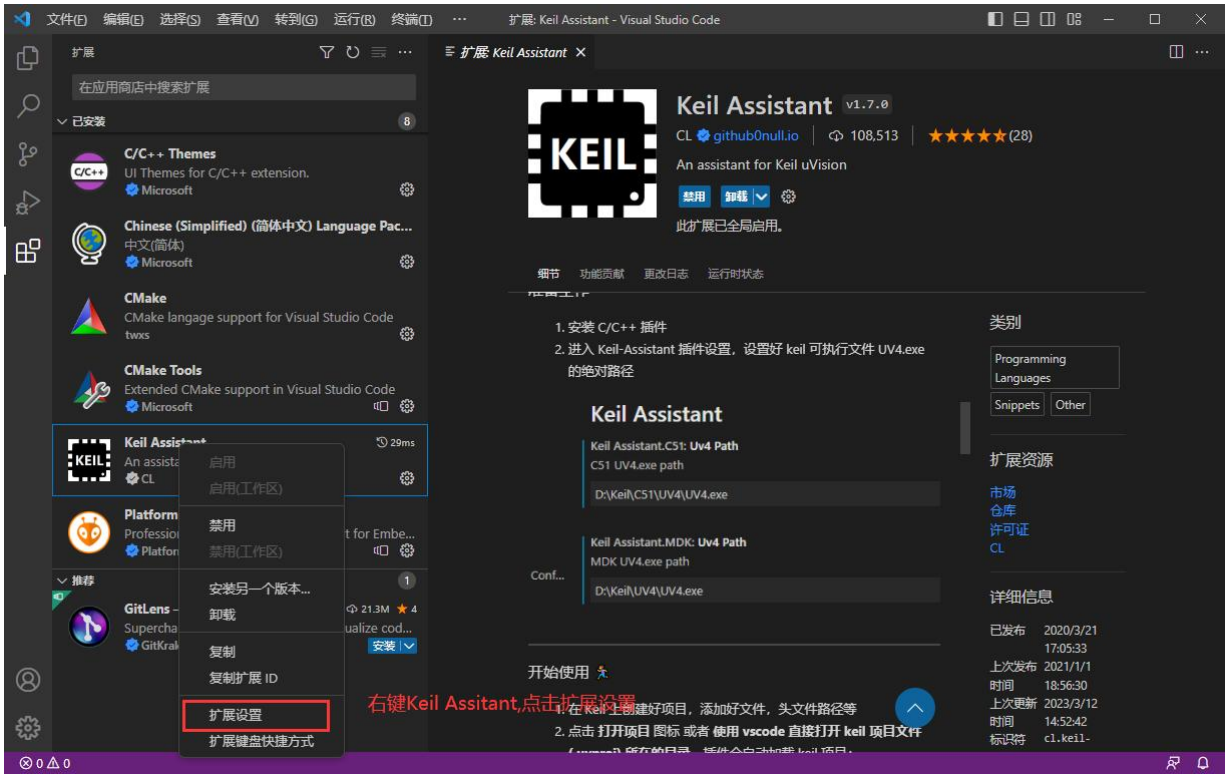


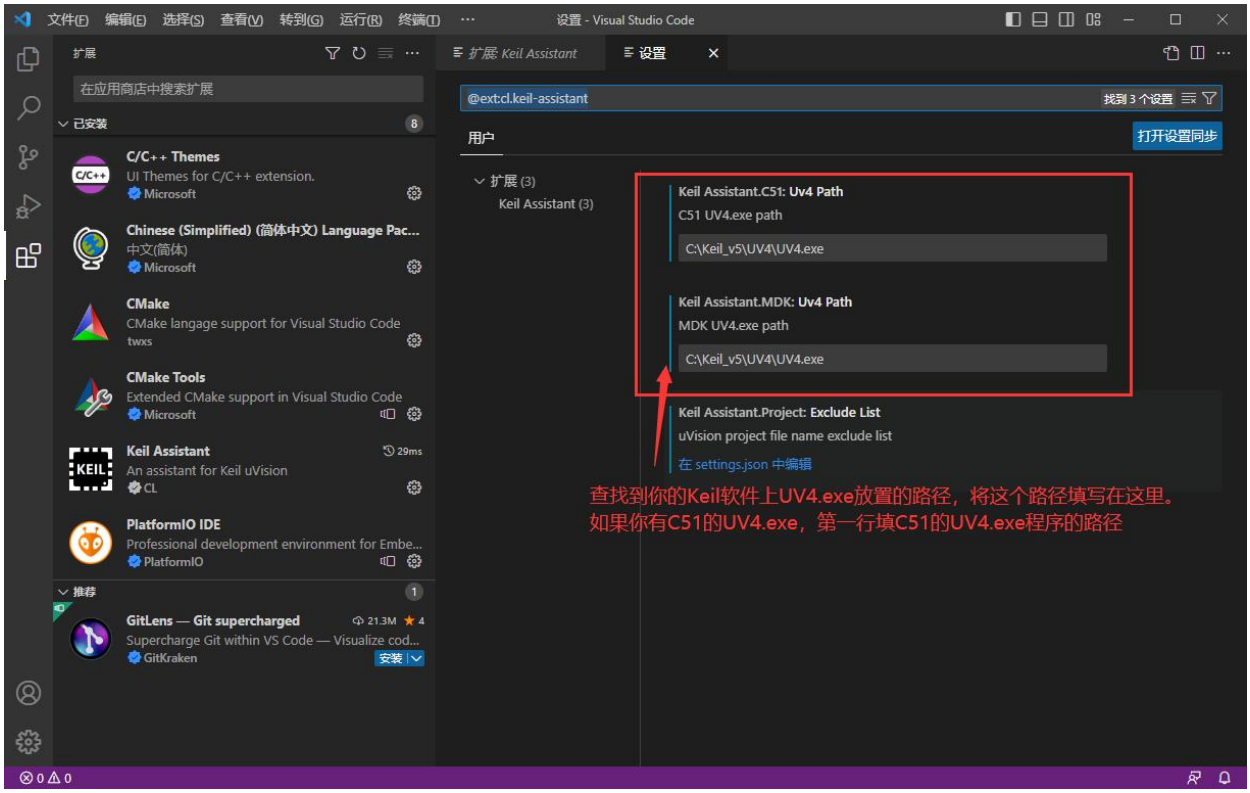
2.3 安装 Keil Assitant

安装 Keil Assitant,要详细阅读自带的使用说明。安装之后还要做一些关于这个扩展的设置才行。



进入 Keil Assitant 的扩展设置界面，需要你填写你已经安装好的 Keil 程序的安装路径，记得要填你自己安装 Keil 程序的安装路径，因为到时候你要编译程序其实是 VSCODE 调用 Keil 的编译功能实现在 VSCOE 上编译工程文档的。



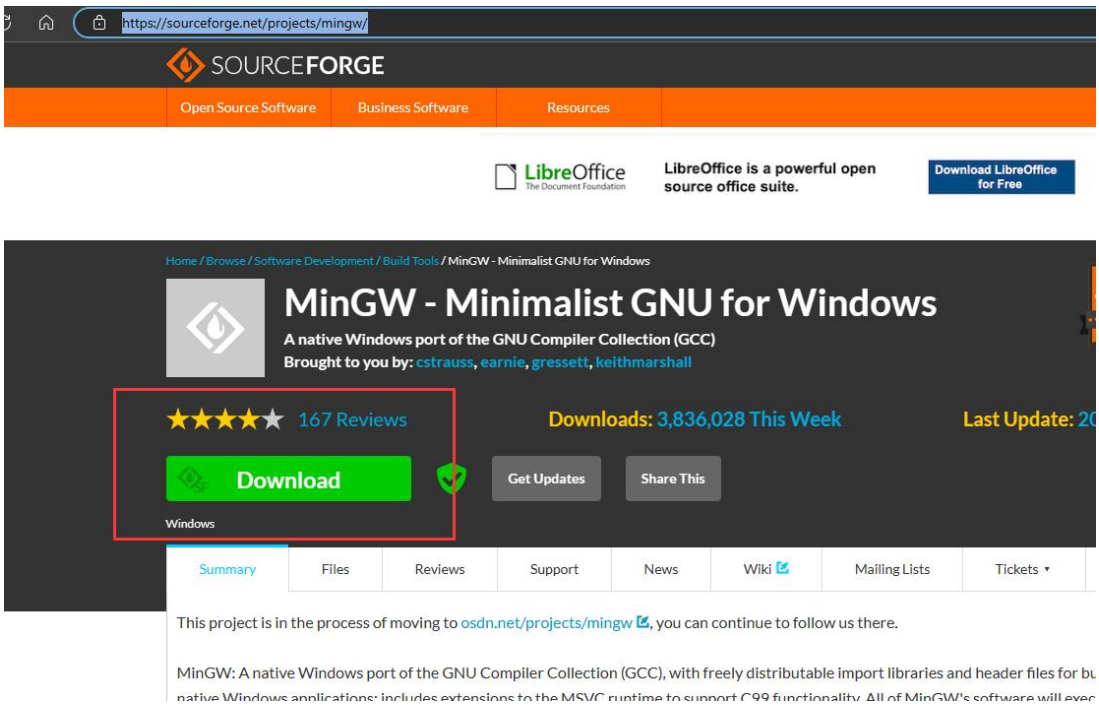


3、安装 MinGW

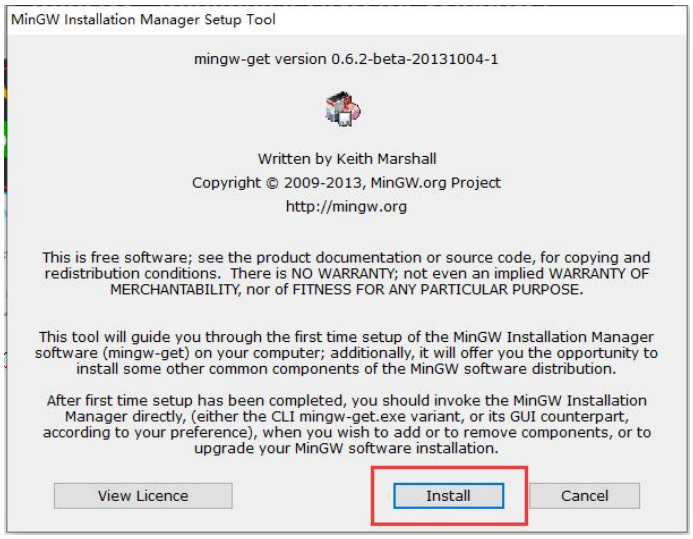
VS Code 是一个代码编辑器，它本身不包含编译器和调试器，所以你需要安装一个外部的编译器和调试器来编译和运行 C 语言代码。MinGW 是一个常用的 Windows 平台的 C++编译器和调试器，所以想要进行编译 C 语言项目的话就必须装这个。

针对 windows PC 用户，你可以从下面的这个网址上下载 mingw-get-setup .exe 这个软件来配置。

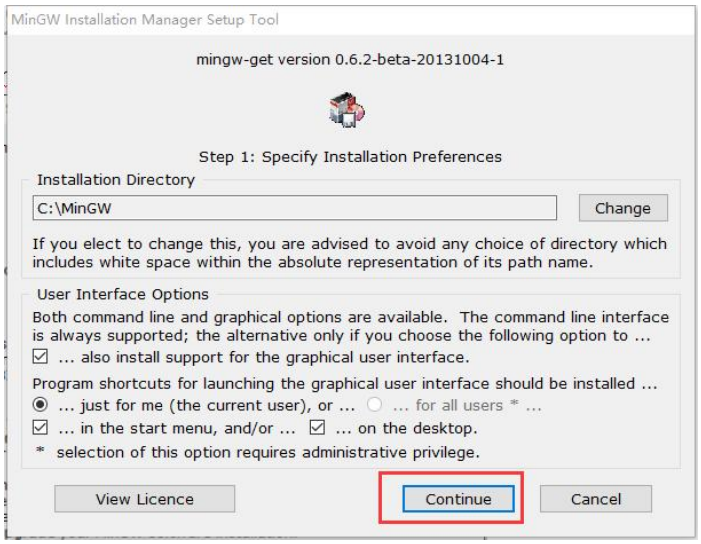
网址：[MinGW - Minimalist GNU for Windows download | SourceForge.net]



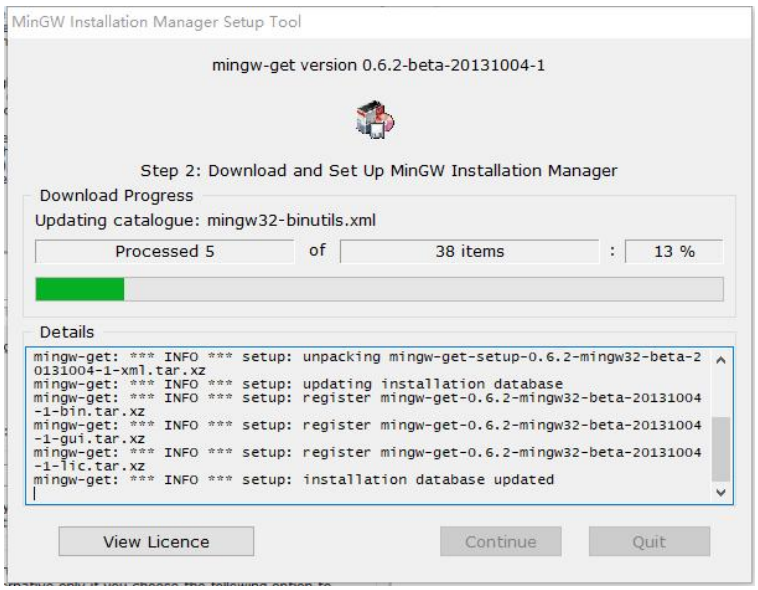
下载成功后点击 mingw-get-setup .exe 这个程序，点击安装



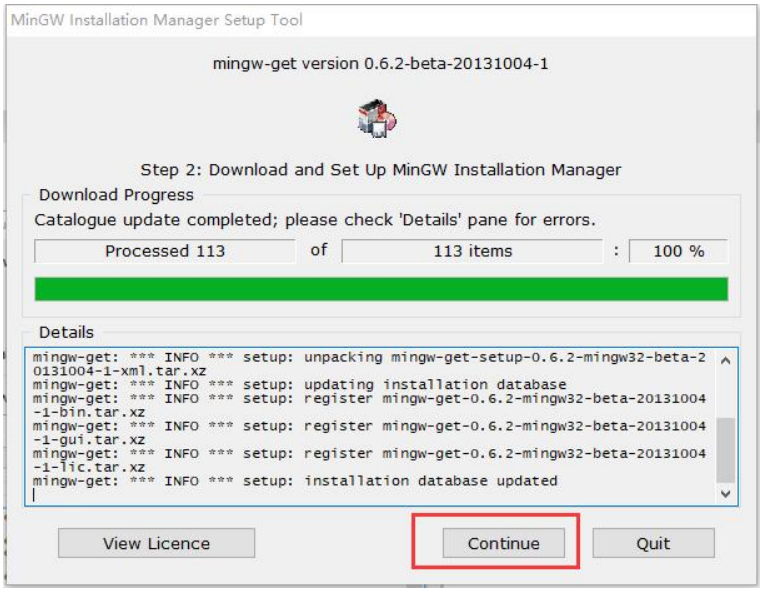
点击继续



等待安装



百分之百后按继续



MinGW Installation Manager 是一个用来管理 MinGW 的安装包的工具，你可以用它来选择和下载你需要的编译器和库。

如果你想配置 C 语言环境，你至少需要安装以下几个 Package：

mingw-developer-toolkit：包含一些开发工具，如 make，gdb 等。

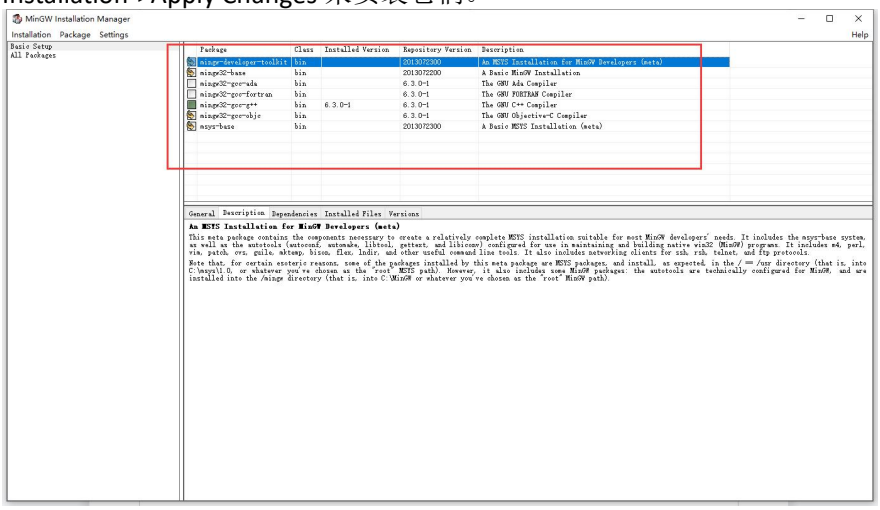
mingw32-base：包含基本的 MinGW 运行时和头文件。

mingw32-gcc-g++：包含 C++编译器。

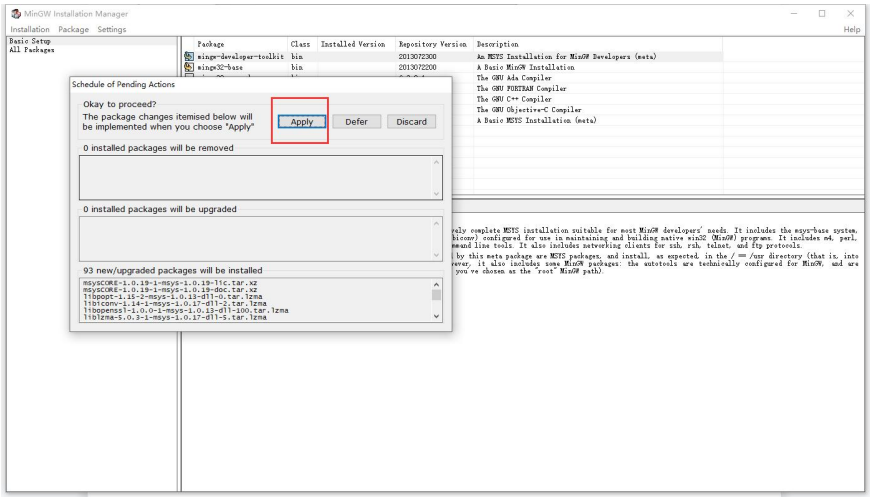
mingw32-gcc-objc：包含 Objective-C 编译器。

msys-base：包含一个类似 Unix 的命令行环境。

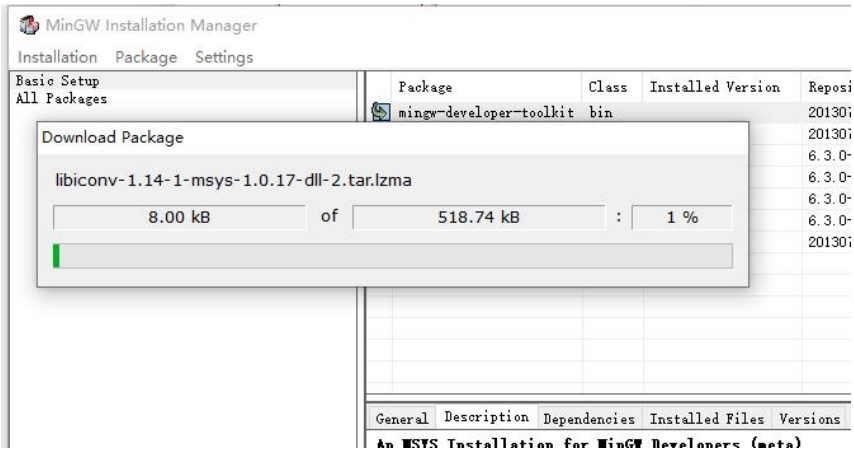
你可以在 MinGW Installation Manager 的 Basic Setup 中找到这些 Package，并且把它们都打上勾，然后点击 Installation->Apply Changes 来安装它们。



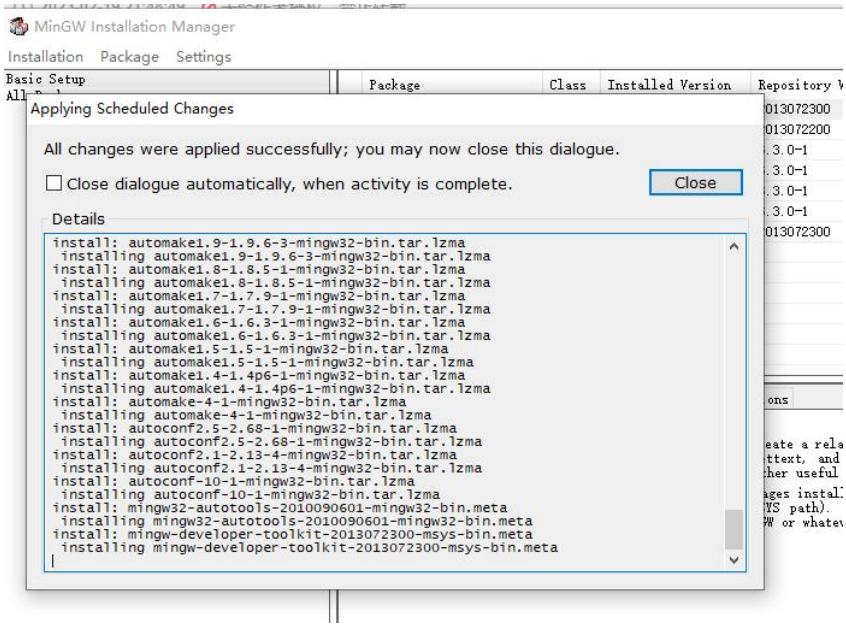
点击 Installation->点击 Apply->安装



等待所有项目完成

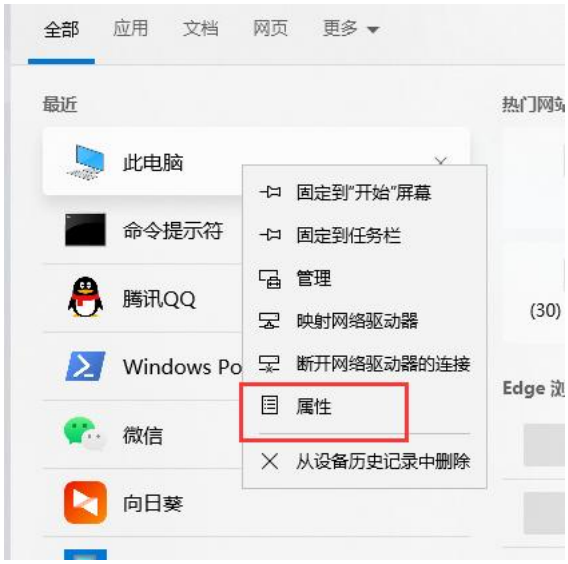


下载完之后点 Close



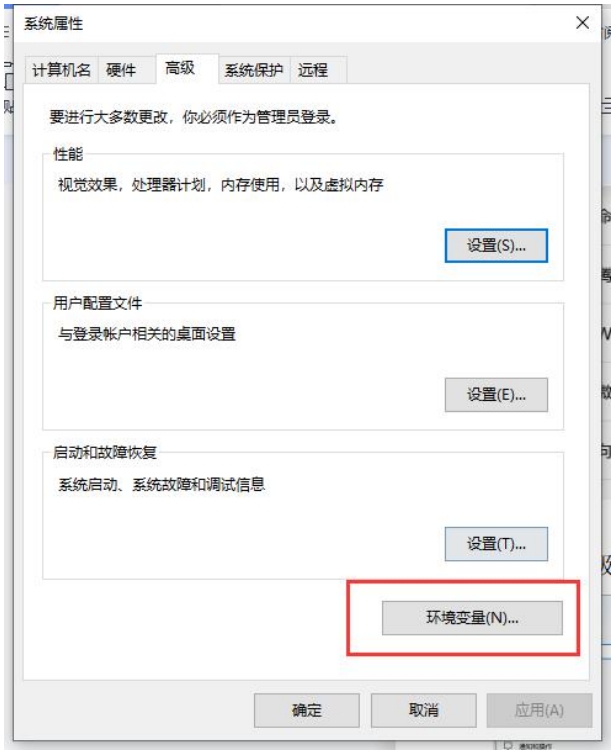
4、配置 MinGW 的环境路径

此时 MinGW 还没配置完全，还需要在 PC 上配置路径。
打开此电脑，右键找到属性点击进去。

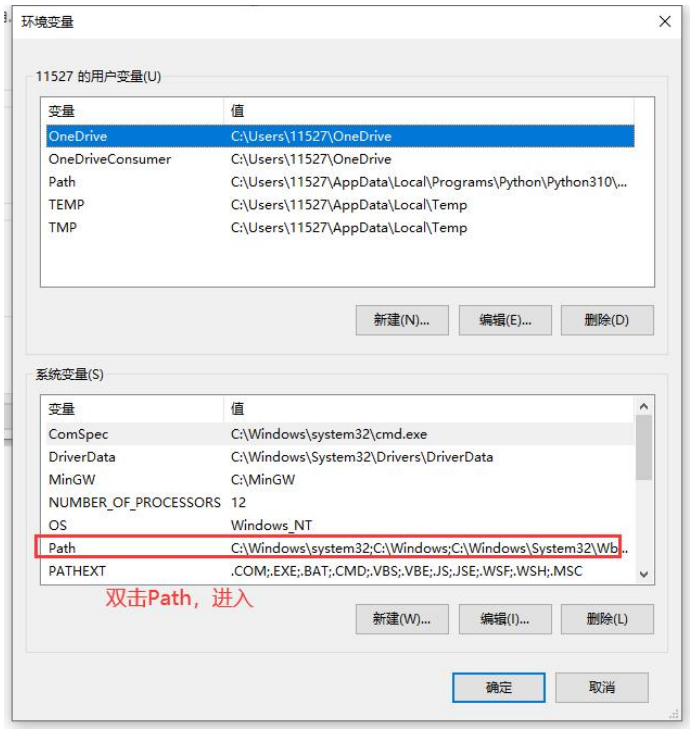


点击“高级属性设置”

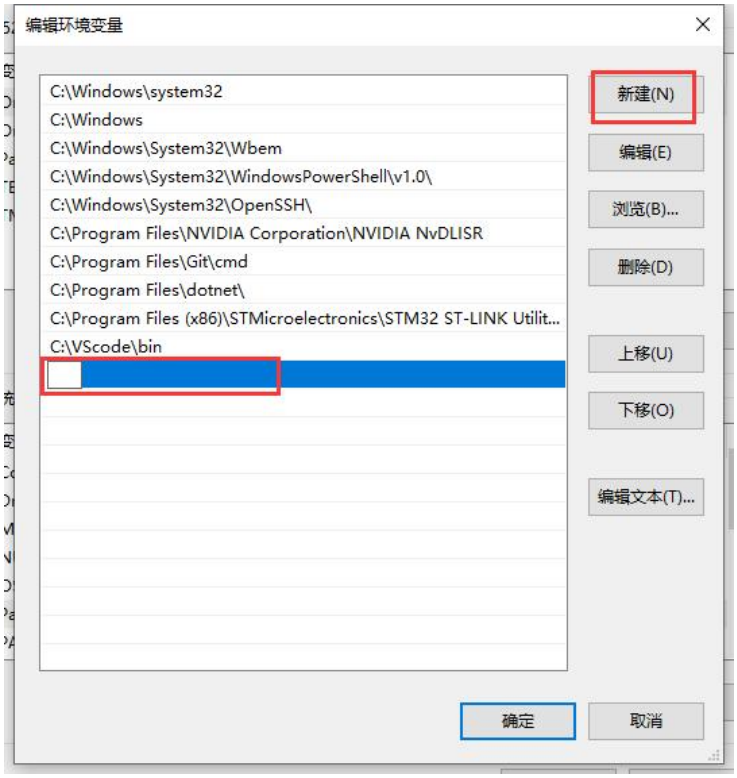
点击环境变量

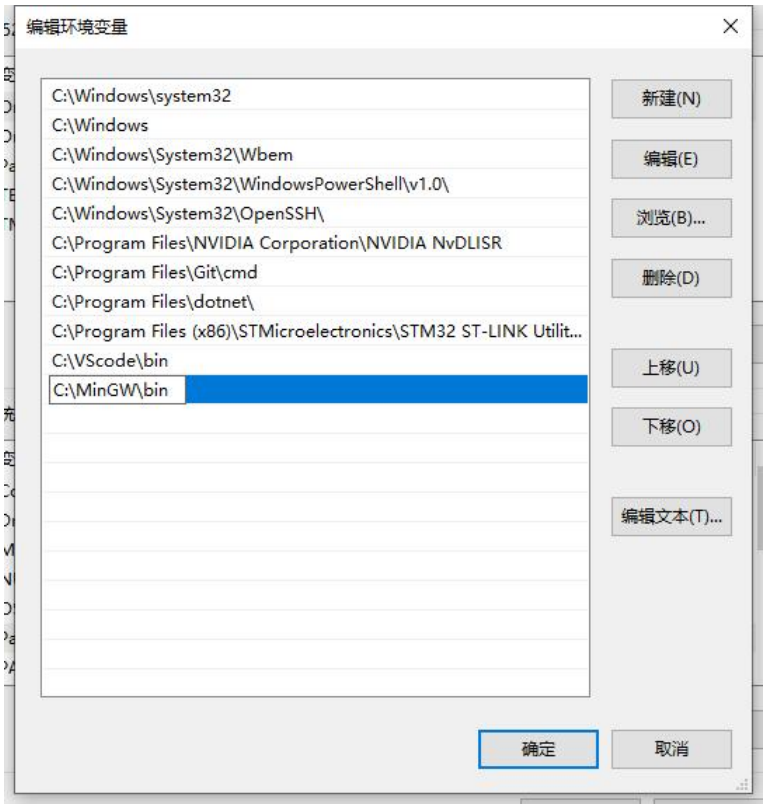


点击系统变量中的“Path”双击

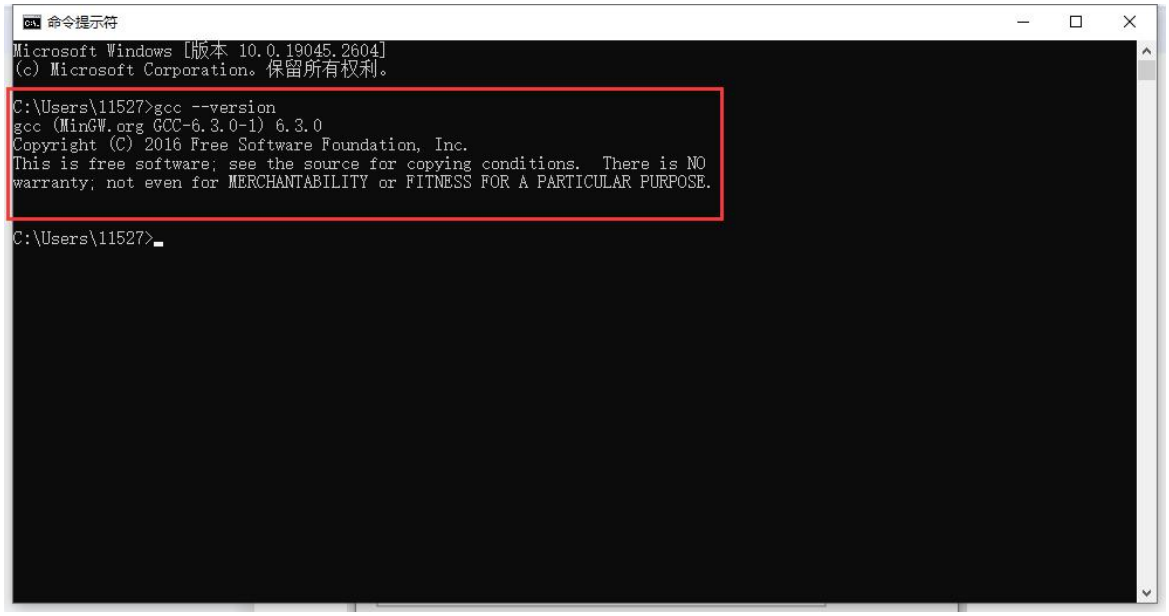


点击新建，找到 MinGW 的路径填进新建的新行里



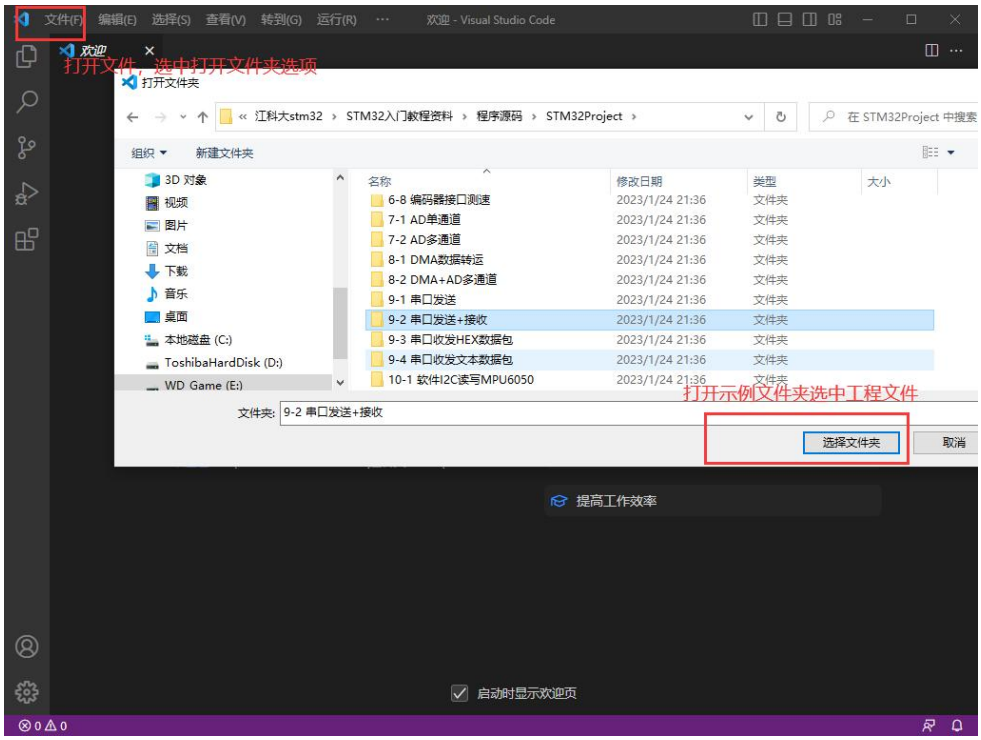


打开 CMD 命令行，输入 gcc --version,查看是否正常安装上 GCC。如下图，可以看到 gcc 的版本号，说明安装成功了。

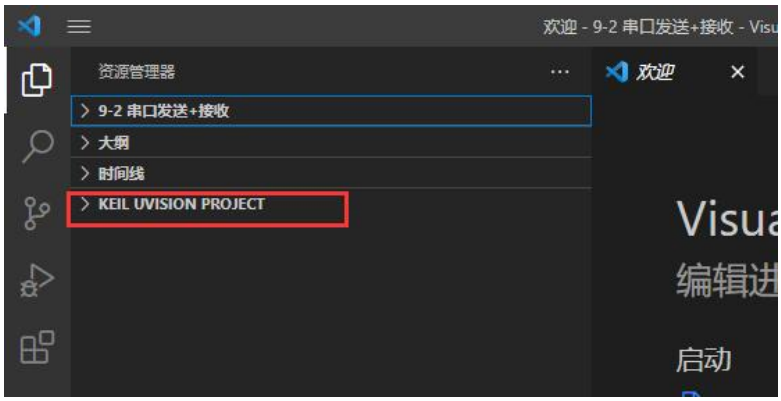


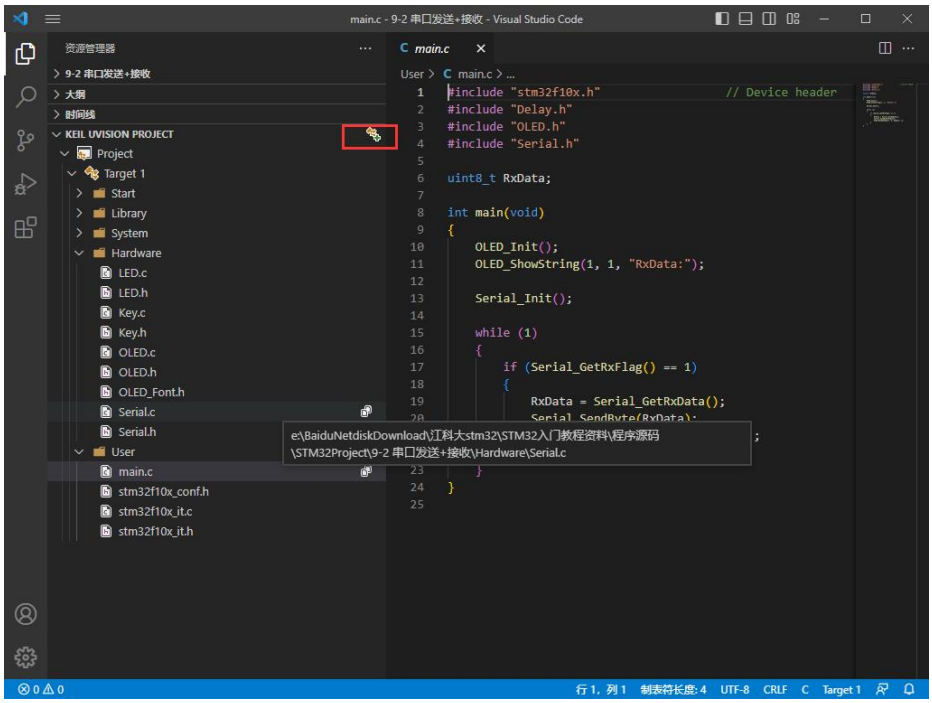
5 打开 Keil 项目

打开 VSCODE,文件 -> 打开文件夹 -> “示例工程文件夹”，选择文件夹并打开

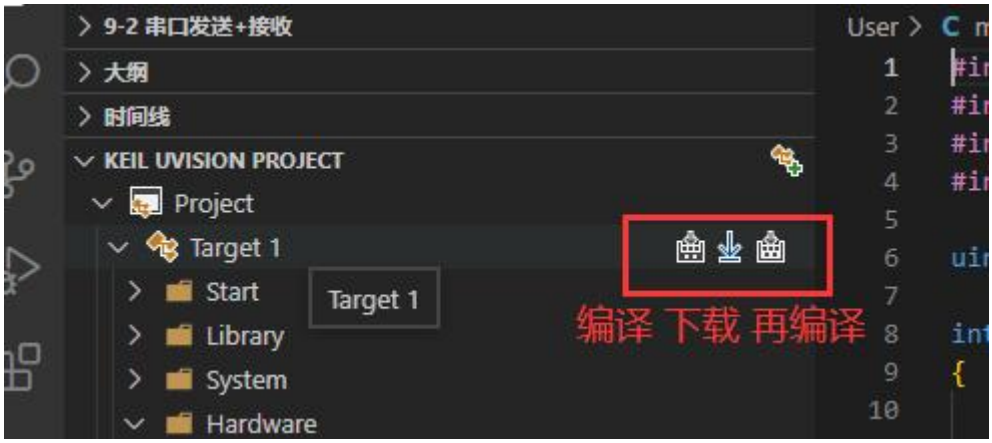


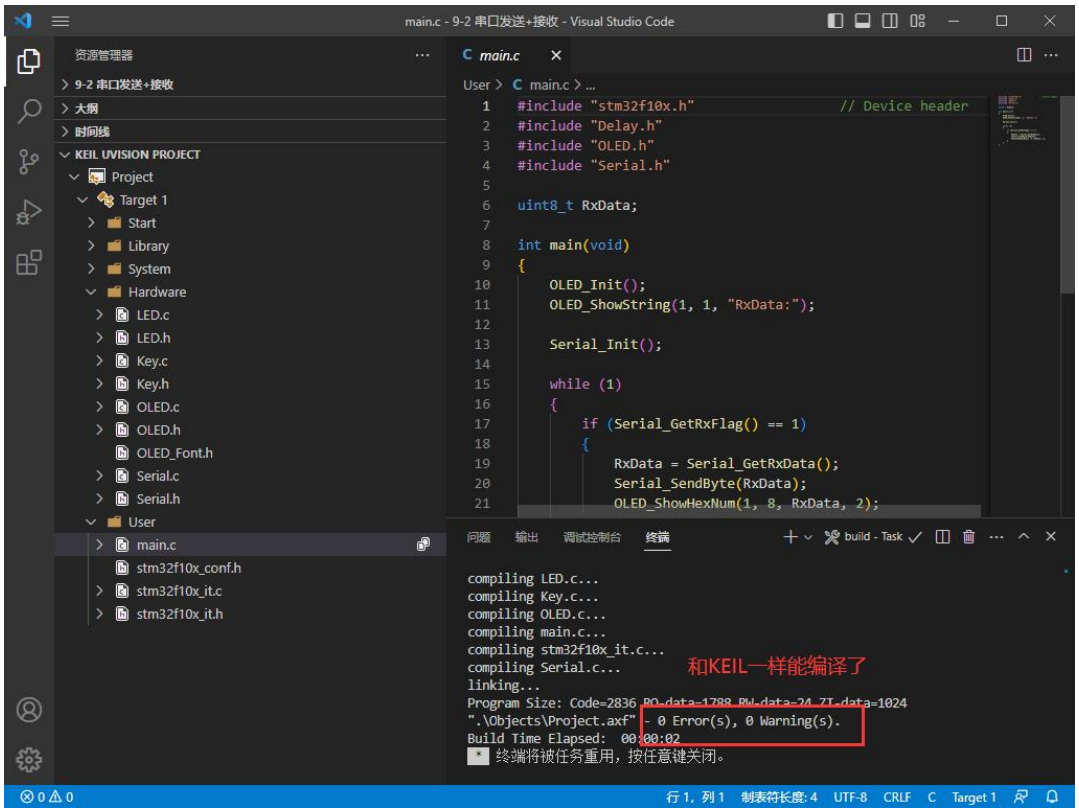
点击 KEIL UVSION PROJECT





此时已经可以正常编译了

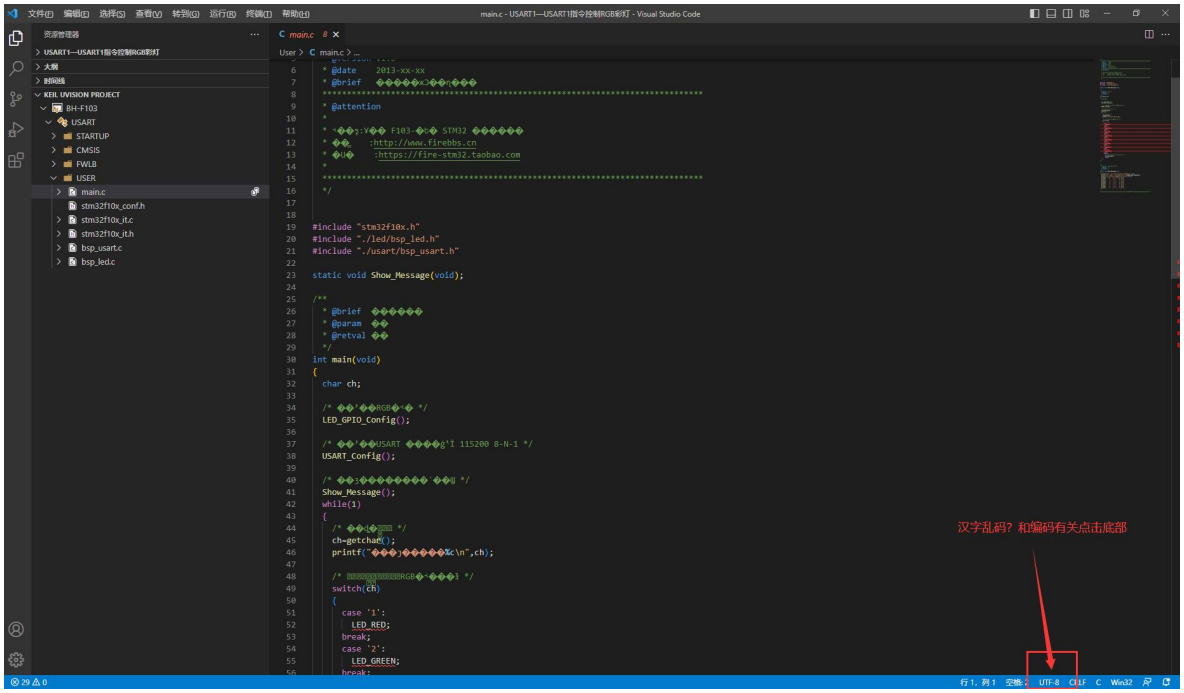




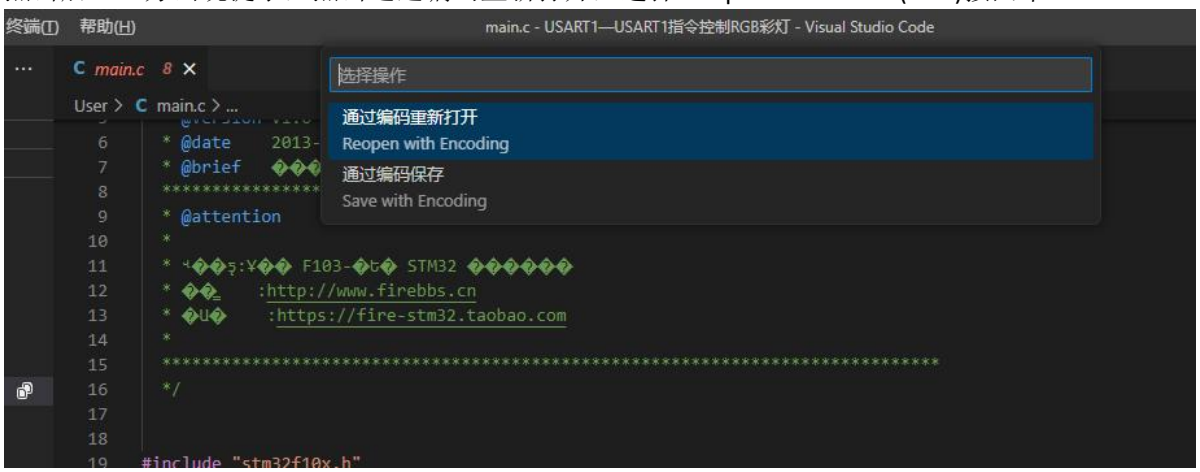
6 可能会遇到的问题

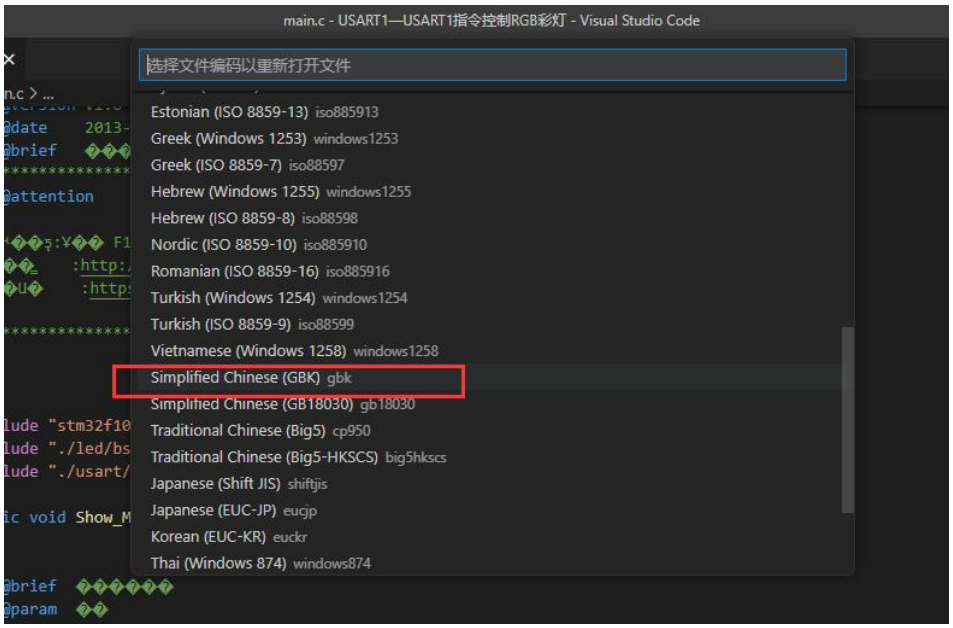
6.1 汉字注释乱码问题：

解决办法：最底部的蓝条点击“UTF-8”

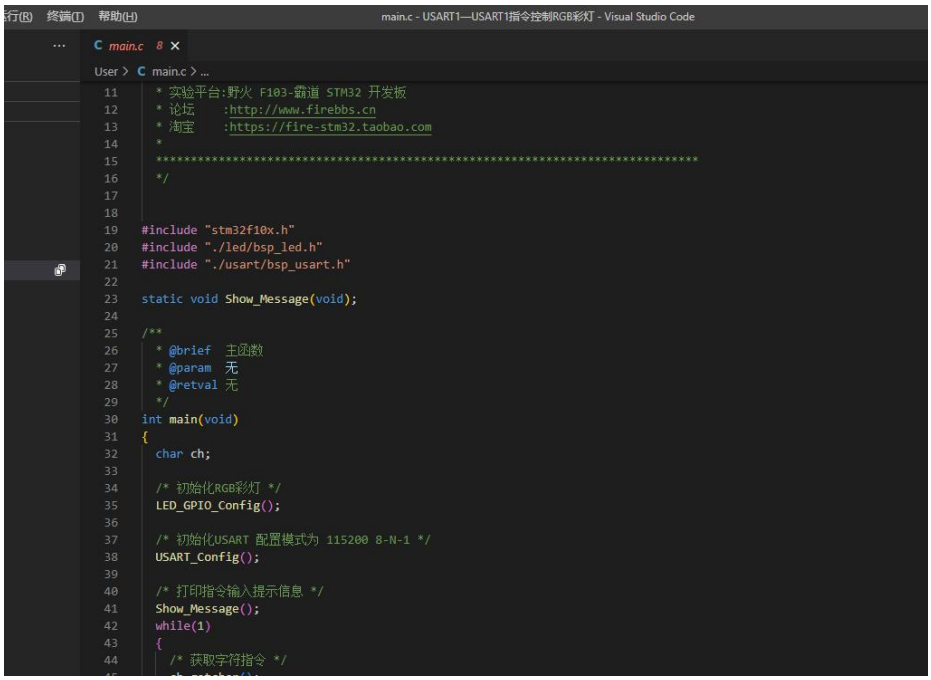


点击后正上方出现提示，点击通过编码重新打开，选择 Simplified Chinese (GBK)按回车。





此时，汉字能正常显示了。



6.2 某些关键词底下出现了报错的红色波浪线怎么办：

我打开野火的示例代码会发现一些库函数的关键词，库函数名词底下会出现报错的红色波浪线，这中问题和#include “xx.h”头文件的缺失有关。KEIL 可以通过魔术棒设置 C/C++项目下的 include path 解决，但是 VSCODE 你需要把波浪线所包含的.h 文件写到当前文件。


```
15      ******************************************************************************
16      */
17
18    #include "bsp_usart.h"
19    #include "stm32f10x_gpio.h"
20    #include "stm32f10x_usart.h"
21    #include "stm32f10x_rcc.h"
22
23    /**
24     * @brief  USART GPIO 配置,工作参数配置
25     * @param  无
26     * @retval 无
27     */
28    void USART_Config(void)
29    {
30        GPIO_InitTypeDef GPIO_InitStructure;
31        USART_InitTypeDef USART_InitStructure;
32
33        // 打开串口GPIO的时钟
34        DEBUG_USART_GPIO_APBxClkCmd(DEBUG_USART_GPIO_CLK, ENABLE);
35
36        // 打开串口外设的时钟
37        DEBUG_USART_APBxClkCmd(DEBUG_USART_CLK, ENABLE);
38
39        // 将USART Tx的GPIO配置为推挽复用模式
40        GPIO_InitStructure.GPIO_Pin = DEBUG_USART_TX_GPIO_PIN;
41        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
42        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
43        GPIO_Init(DEBUG_USART_TX_GPIO_PORT, &GPIO_InitStructure);
44
45      // 将USART Rx的GPIO配置为浮空输入模式
46        GPIO_InitStructure.GPIO_Pin = DEBUG_USART_RX_GPIO_PIN;
47        GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
48        GPIO_Init(DEBUG_USART_RX_GPIO_PORT, &GPIO_InitStructure);
49
50        // 配置串口的工作参数
51        // 配置波特率
52        USART_InitStructure.USART_BaudRate = DEBUG_USART_BAUDRATE;
53        // 配置 针数据字长
54        USART_InitStructure.USART_WordLength = USART_WordLength_8b;
55        // 配置停止位
56        USART_InitStructure.USART_StopBits = USART_StopBits_1;
57        // 配置校验位
58        USART_InitStructure.USART_Parity = USART_Parity_No ;
59        // 配置硬件流控制
60        USART_InitStructure.USART_HardwareFlowControl =
61        USART_HardwareFlowControl_None;
62        // 配置工作模式，收发一起
63        USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
```

6.3 虽然下滑波浪线问题解决了，但是编译功能又不能用了？

打开某些示例文档，发现编译功能不能用了。这时候可能和工程文档的某些 KEIL 的设置有关，我目前还没找到原因。不过可以选择在 VSCODE 上查看编辑文档，在 VSCODE 上点击保存后再在 KEIL 打开文档进行上编译。两个工具搭配起来一起用。