

AIO工程的开发环境搭建教程

——by @神秘藏宝室、@无常喵态

2022/04/29

0.前言

1.本教程适用于Ver2.0.5及之前版本的AIO工程，之后版本是否适用，取决于固件作者 [溜马小哥](#) 是否会更新Espressif32框架。

2.AIO工程基于Espressif32框架下的Arduino框架，在Visual Studio Code插件PlatformIO IDE下进行开发。

3.由于近期Espressif32框架跨大版本更新为Ver4.1.0，其下的Arduino框架也更新为Ver2.0，新旧框架不能兼容，故若要使用AIO工程，搭建环境时必须使用与AIO工程相匹配的Ver3.5.0，否则编译失败。下文会有详细说明。

4.因为众所周知的原因，搭建过程中某些步骤需要科学上网，所以本教程也提供了名为“PIO_ESP32离线包.zip”的压缩文件（下文统称“离线包”），位于群文件“面向开发者工具”文件夹，大家按需自取，教程中也会介绍该离线包的使用方法。

1.准备工作

安装PlatformIO IDE前，要先安装好Python和VS Code。

1.1 安装 Python

确认自己电脑装有3.6及更高版本的Python，且已经配置好了环境变量（Python环境变量的配置请自行百度，此处不再赘述）。

下方链接为Python 3.10.4版本的下载地址，点击即可弹出下载窗口，安装过程中记得勾选“将Python加入环境变量”。

<https://www.python.org/ftp/python/3.10.4/python-3.10.4-amd64.exe>

1.2 安装 Visual Studio Code

VS Code的安装无注意事项，正常安装即可。

下方链接为VS Code 1.66.2版本的下载地址，点击即可弹出下载窗口。

<https://az764295.vo.msecnd.net/stable/dfd34e8260c270da74b5c2d86d61aee4b6d56977/VSCodeSetup-x64-1.66.2.exe>

2.安装PlatformIO IDE

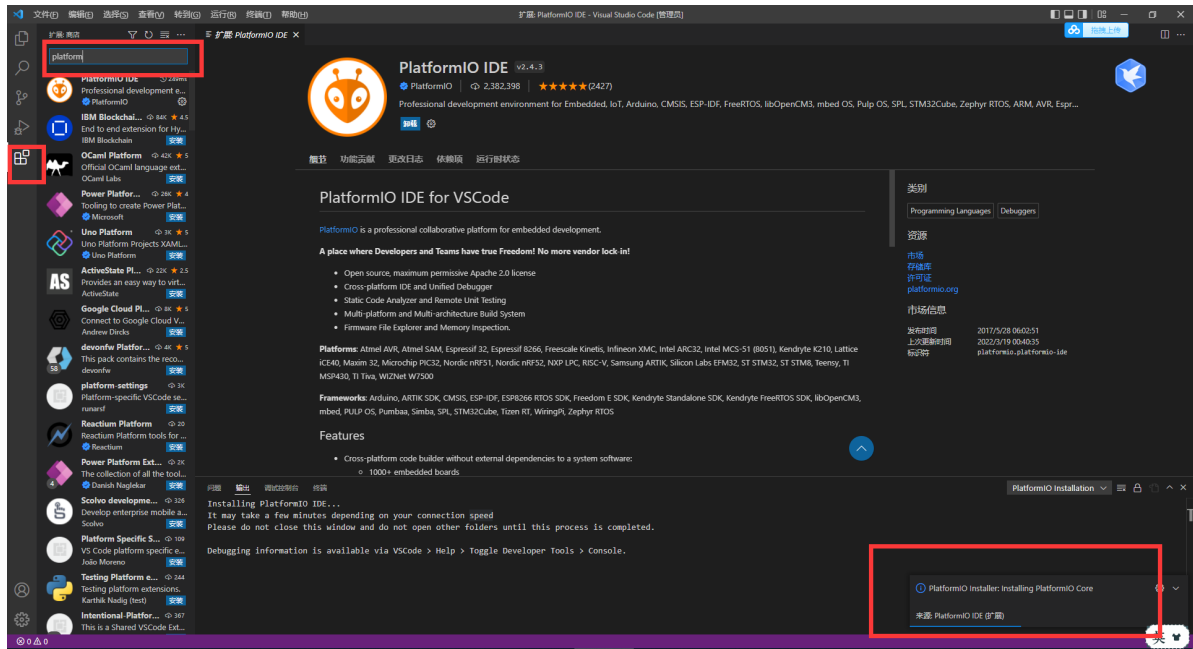
此处提供两种方式：等待式安装和换源命令式安装，两种方式最终效果相同，二选一即可。

无脑等待式安装适合小白，只需三两步操作，然后等待安装完成即可；缺点是，若网络条件欠佳且不会科学上网，会半途卡三天三夜（丝毫不夸张），且有可能陷入卡住死循环。

换源命令式安装适合有一定命令行基础的群友，通过命令行操作脚本，将镜像源切换至国内清华大学镜像库，可以有效缩短安装时间。

2.1 等待式安装

如下图所示，打开VS Code，点击界面左侧红框内的方形按钮，在弹出来的搜索框中输入“Platform”，下方会自动出现黄白色蚂蚁头图案的插件，点击该插件右下角的蓝色安装按钮，会出现界面右下角的进度条提示框。



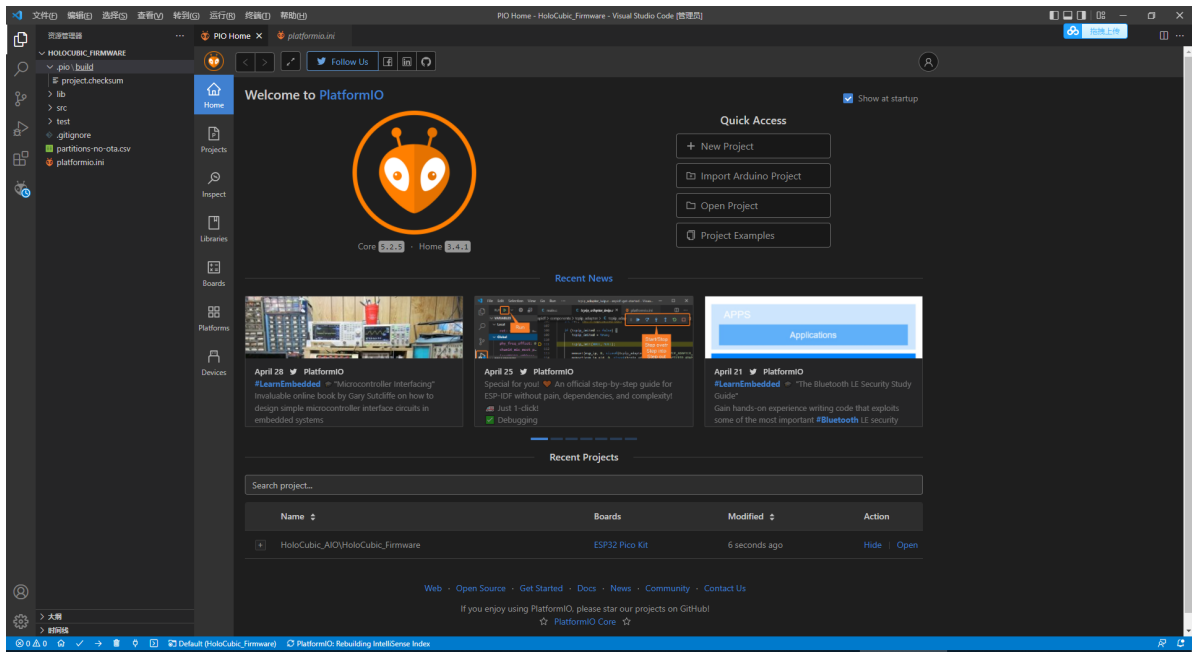
如果中途提示安装Python，选择已经安装过，把python的路径复制进去，比如我的路径是：

```
1 | C:\Users\Administrator\AppData\Local\Programs\Python\Python310\python.exe
```

之后回车。

根据自己的具体地址填写，实在不会可以去环境变量path里面找路径。

等待其自动跑完消失，会出现重加载提示，点击reload，界面左侧出现下图所示的蚂蚁头图标即表示安装完成。



注：这个过程漫长等待，一般要一小时左右，而且需要梯子科学上网，若不想等待可以退出，选择 2.2 换源命令式安装。

2.2 换源命令式安装

1) 删除残留目录文件

首先如果有以下2个文件夹请先删除：

`C:\Users\（你的用户名）\.platformio`

`C:\Users\（你的用户名）\.vscode\extensions\platformio.platformio-ide-x.x.x`


2) 更换镜像源为清华镜像站

①新建一个如下内容的 `pip.ini` 的文件并保存：

```
1 [global]
2
3 index-url = https://pypi.tuna.tsinghua.edu.cn/simple
```

②将上一步中创建的 `pip.ini` 文件移动到如下图所示的路径中（将下图路径中的“Administer”替换为你自己的用户名）。

共享 查看

此电脑 > Win 10 Pro x64 (C:) > 用户 > Administrator > pip			
名称	修改日期	类型	大小
 pip.ini	2022/4/28 23:21	配置设置	1 KB

注：若不会创建 `pip.ini` 文件，可将“离线包”内“pip”文件夹中的文件复制到上图所示路径。

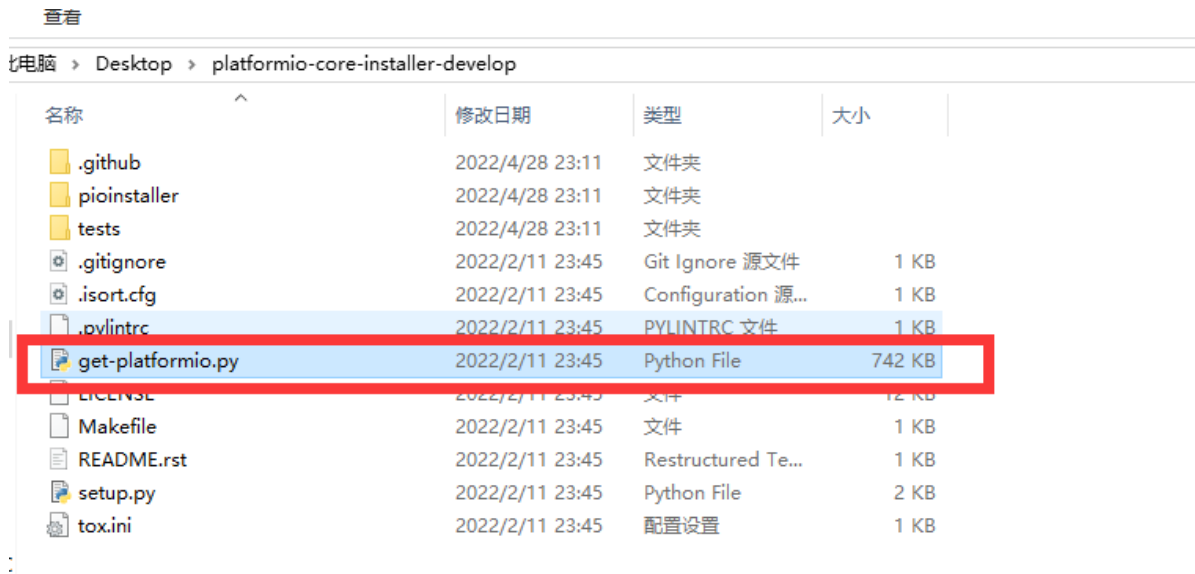
3) 安装Pio-core

①点击下方链接，下载PIO-core的安装文件。

<https://github.com/platformio/platformio-core-installer/>

注：若科学上网有困难，可使用“离线包”中 2. 安装PlatformIO IDE 文件夹中的 platformio-core-installer-develop.zip 文件，直接进入下一步。

②解压上一步下载的压缩包，双击其中下图所示的 get-platformio.py 文件；

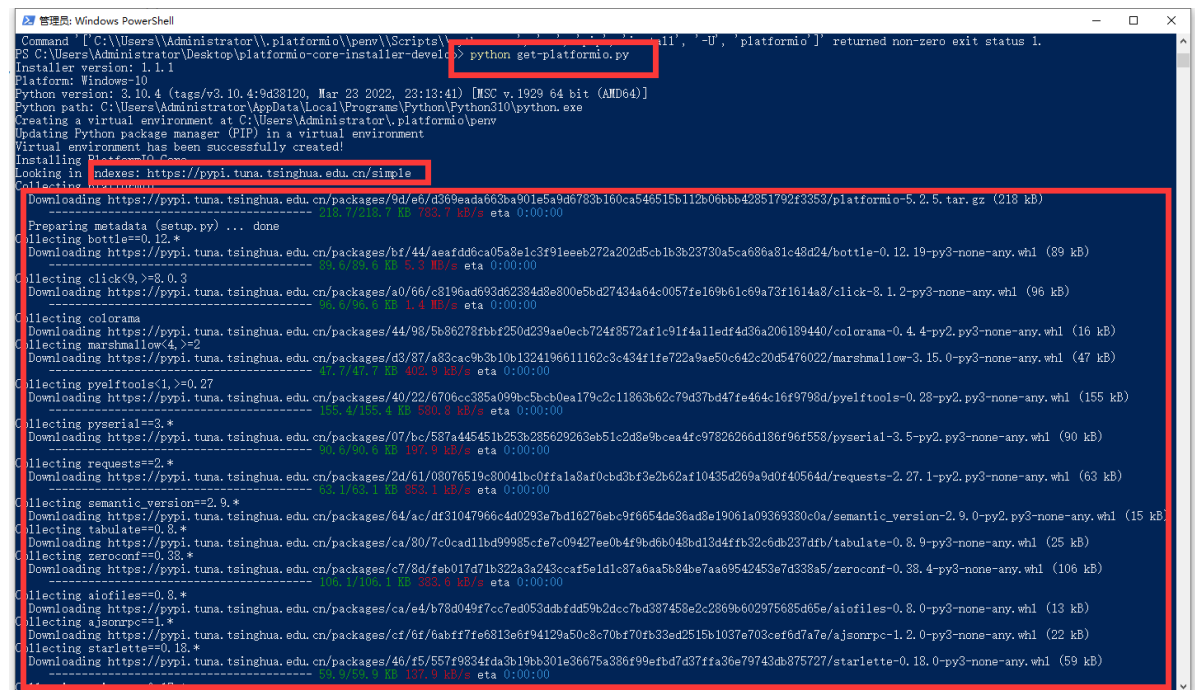


或者用python指令安装；

或者在所在文件夹空白处 shift+右键，选择PowerShell指令，输入如下命令安装：

```
1 python get-platformio.py
```

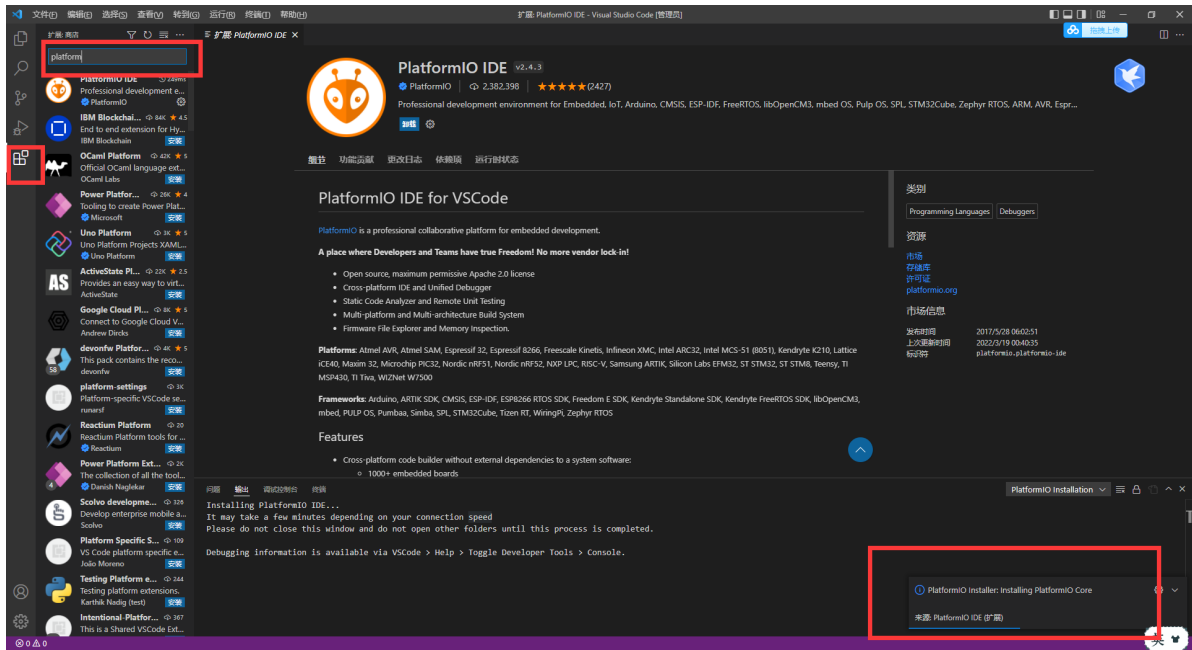
注：此处不要科学上网，否则易出问题。



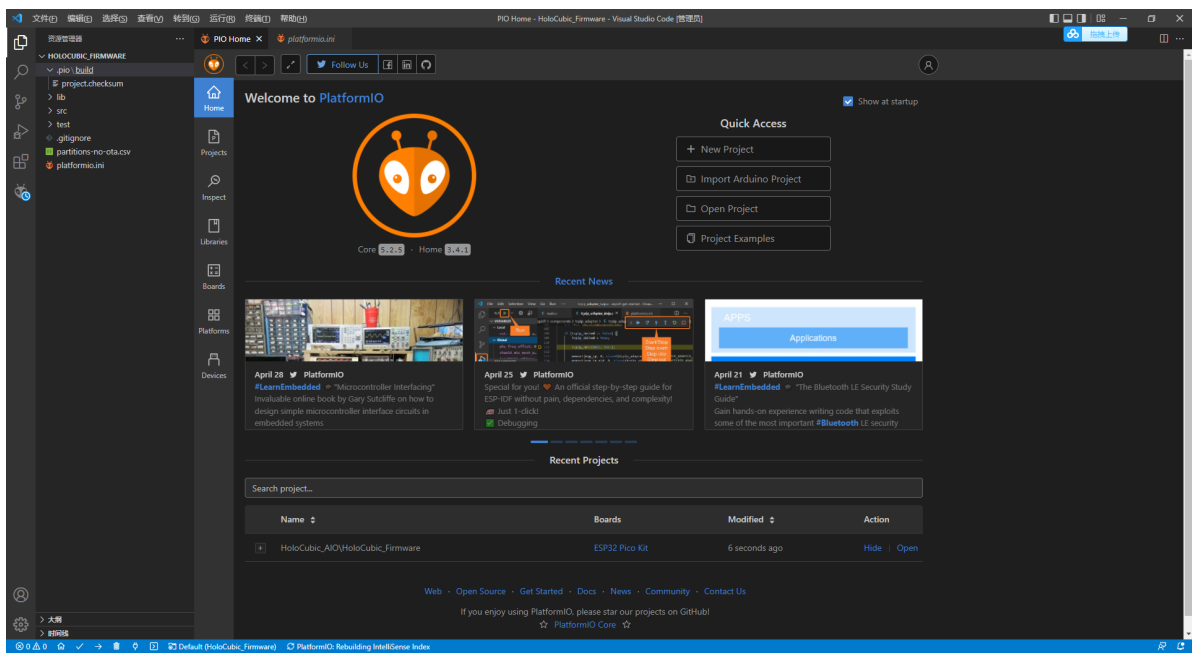
如上图，等待安装完成即可。

4) 安装PlatformIO IDE

如下图所示，打开VS Code，点击界面左侧红框内的方形按钮，在弹出来的搜索框中输入“Platform”，下方会自动出现黄白色蚂蚁头图案的插件，点击该插件右下角的蓝色安装按钮，会出现界面右下角的进度条提示框。



等待约5分钟，进度条跑完消失，会出现重加载提示，点击reload，界面最左侧出现如下图所示的蚂蚁头图标即表示安装完成。



3.配置PlatformIO IDE

如前言所述的框架不兼容问题，故安装完PlatformIO IDE后，不能直接打开和编译AIO源码工程，需要配置框架。

此处提供两种配置方式：在线配置和离线配置，两种方式的最终效果相同，二选一即可。

在线方式会让你清楚过程中做了哪些改变，但根据网络条件的好坏，可能会有较长时间的等待，甚至会无限load task。原因是过程中会下载所需要的Arduino框架和交叉编译工具链。如果网络较差，等待时间上不封顶；若网络较好，几分钟就能搞定。

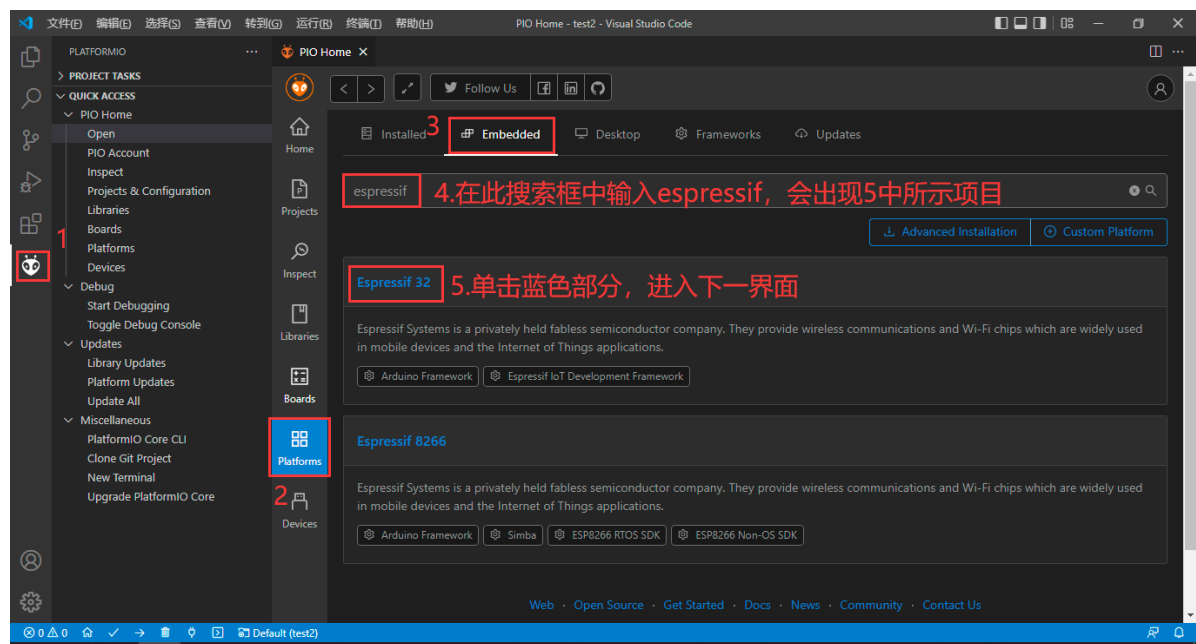
离线方式跳过了在线配置的所有步骤，直接把所需的所有文件一键粘贴到对应路径，然后重启VS Code即可直接进行编译。

推荐使用离线配置方式。

3.1 在线配置

1) 更换框架版本

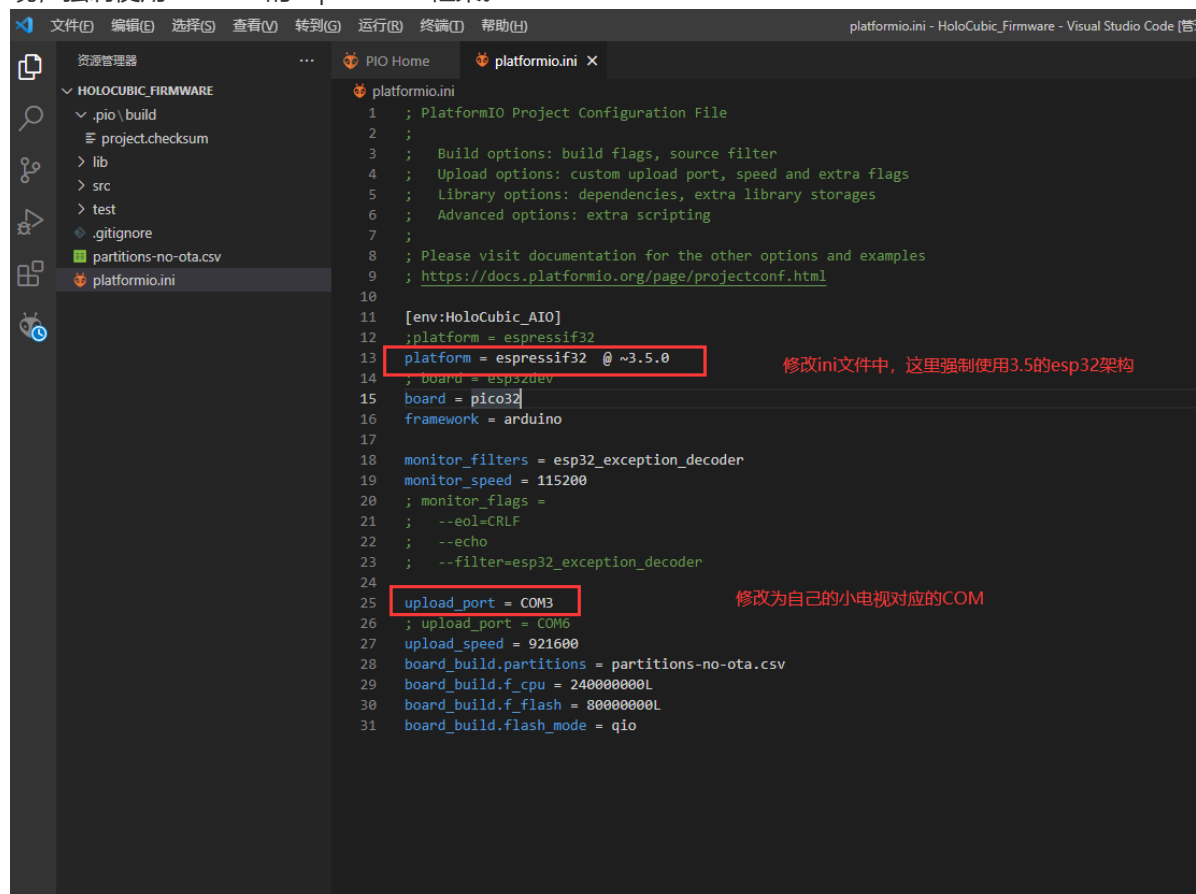
按下图所示进入下一页面



图示“进入下一界面”后，选择Ver3.5.0，并单击蓝色按钮进行安装，等待弹出安装完成的提示框。

2) 限制框架版本

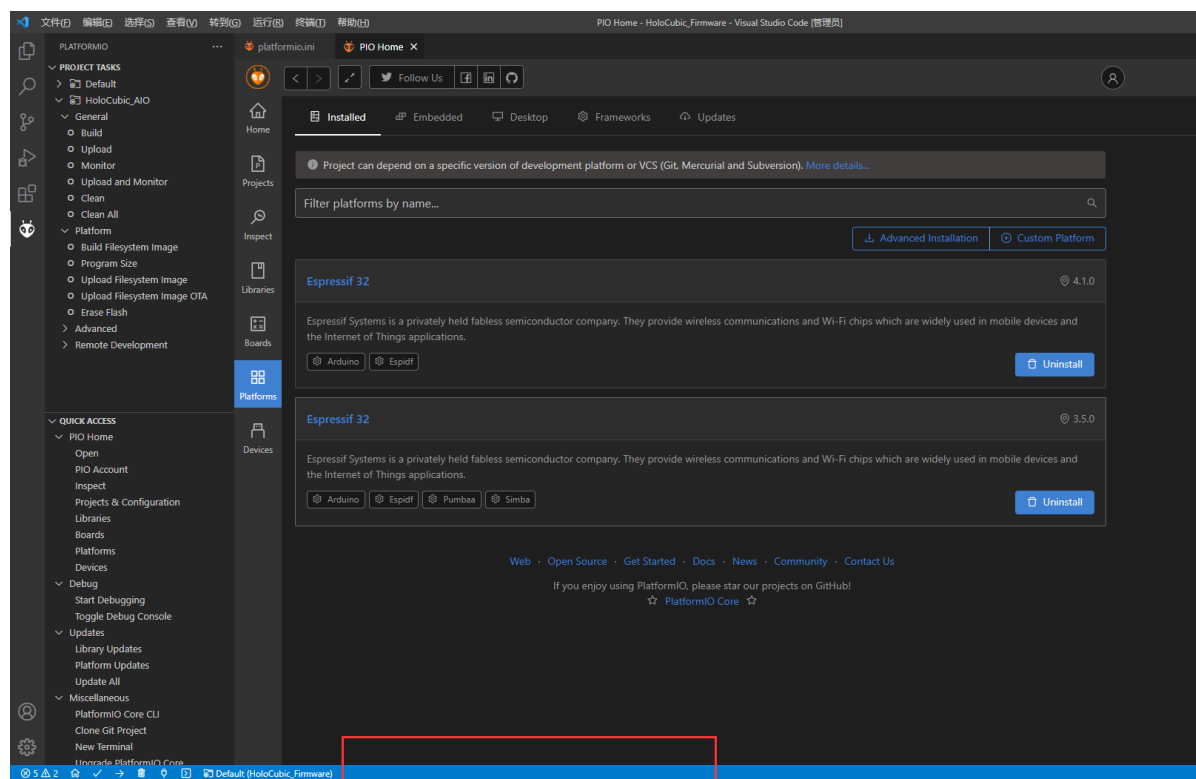
在Platformio IDE中打开AIO工程文件，如下图所示修改 `platformio.ini` 文件中的内容，重启开发环境，强制使用Ver3.5.0的Espressif32框架。



3) Rebuild

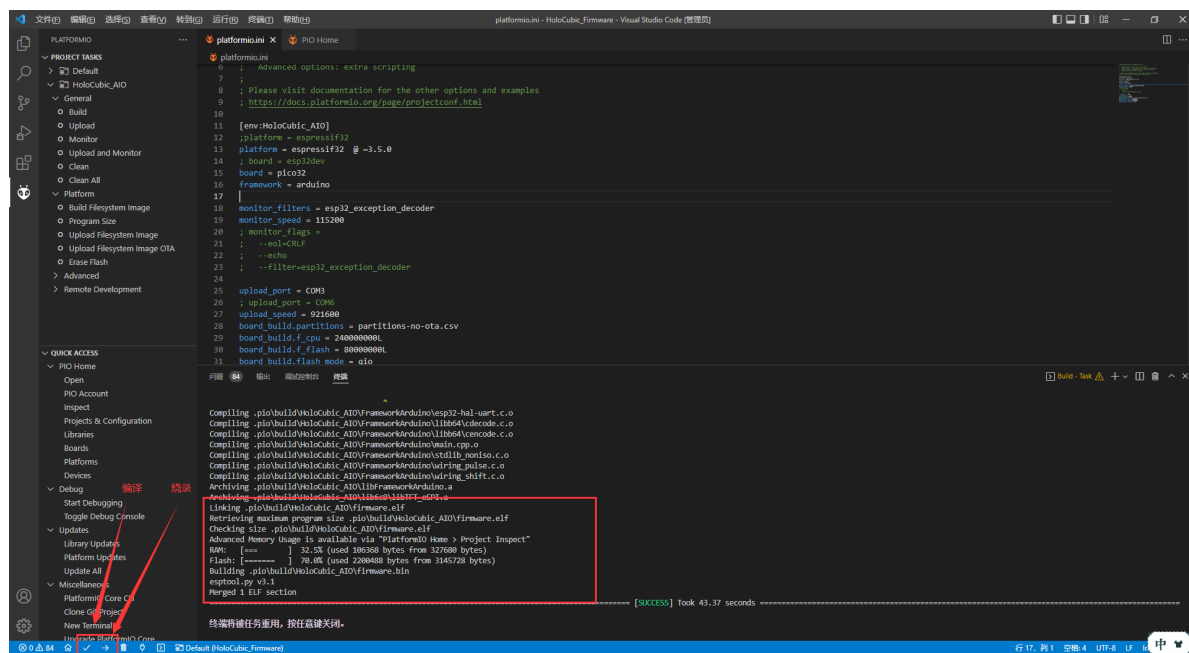
重启后，重新打开AIO工程，下图所示红框内会有 `PlatformIO: Loading tasks...` 字样，等待其消失即可，等待时长据网络条件而定。

注：即使科学上网也不能保证缩短等待时长。



4) 编译和烧录

上一步的字样消失后，即可点击下图所示按钮进行编译和烧录。编译成功后就如下图所示，显示绿色的“success”字样。

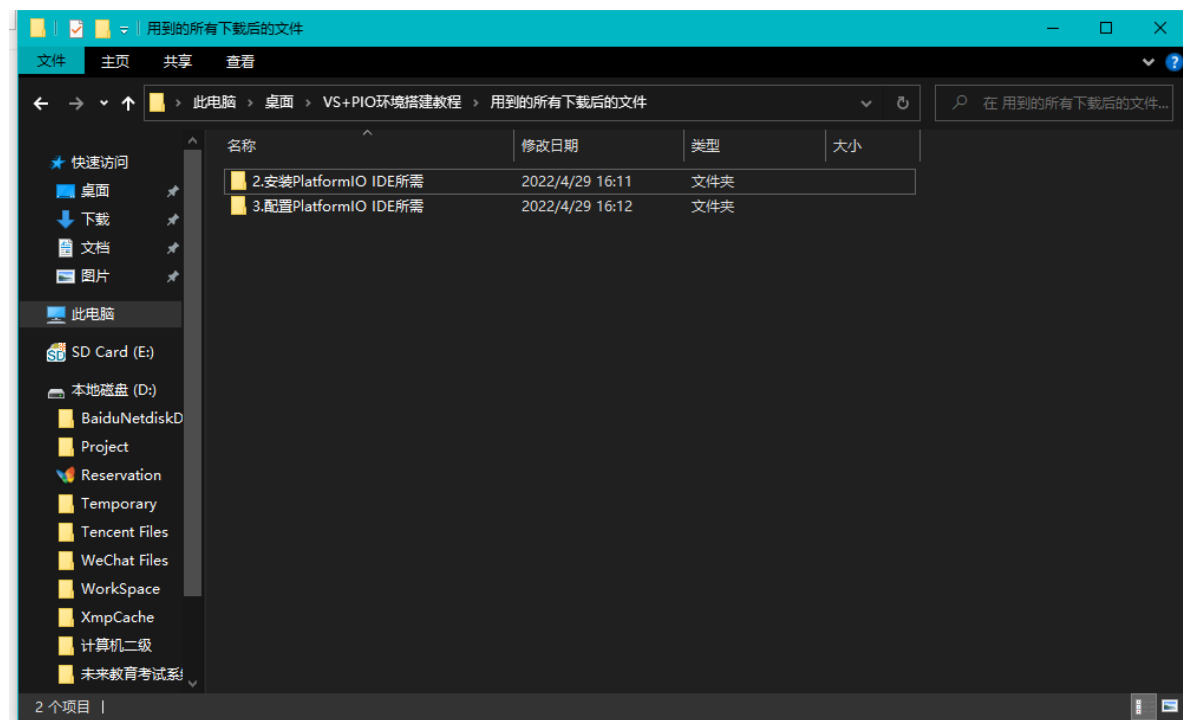


注：如果是首次烧录，根据网络条件，烧录一开始可能会有不定时长且无提示的卡顿，这是因为首次烧录时会在线下载一个烧录所需的工具，等待其下载完毕，最终显示绿色“success”字样即为烧录成功。

3.2 离线配置

1) 替换工具链

下载并解压上文提到过的“离线包”，解压后如下图所示：



将 3. 配置PlatformIO IDE所需 文件夹中名为 packages 的压缩包解压，并将其中所有文件夹“替换粘贴”到如下路径中：

```
1 | C:\Users\（你的用户名）\.platformio\packages
```

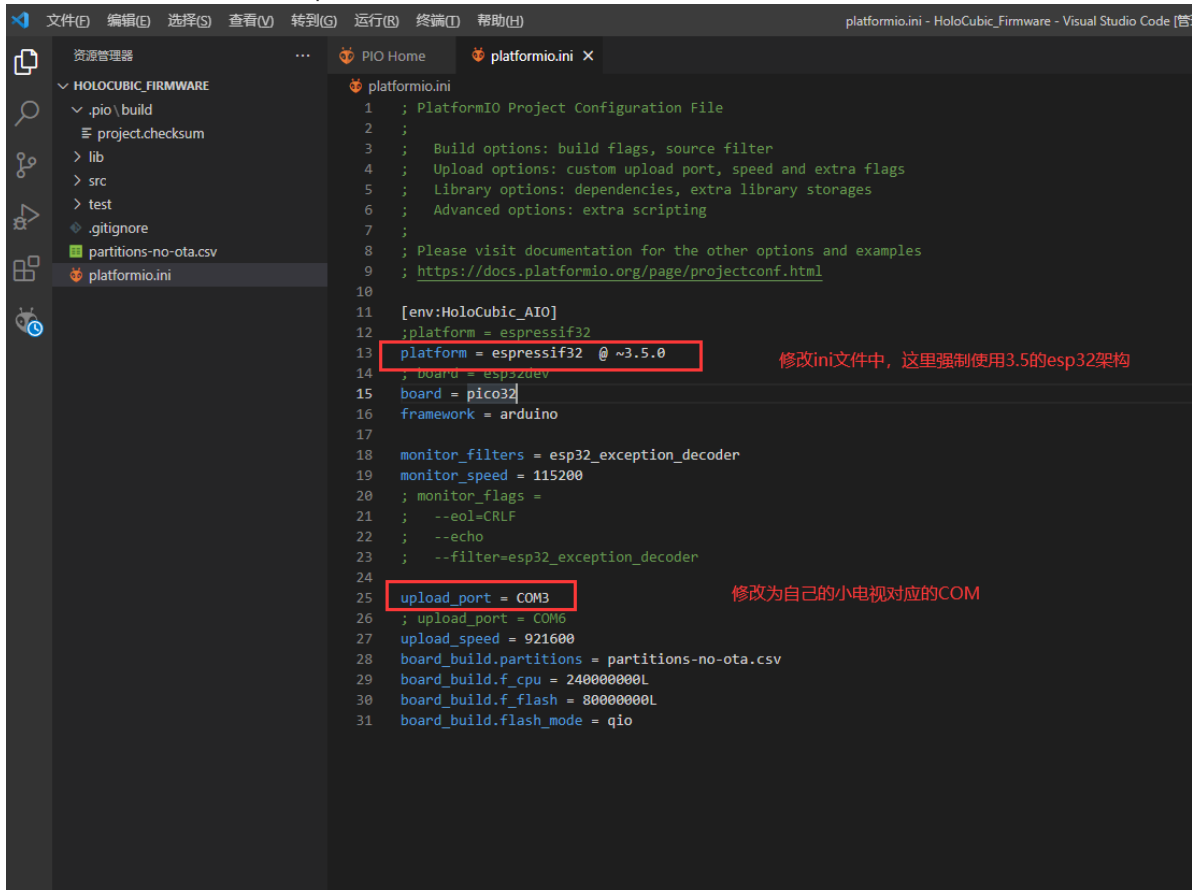
2) 替换框架文件

将 3. 配置 PlatformIO IDE 所需 文件夹中名为 platform 的压缩包解压，并将其中所有文件夹“替换粘贴”到如下路径中：

```
1 | C:\Users\（你的用户名）\.platformio\platforms
```

3) 限制框架版本

在 Platformio IDE 中打开 AIO 工程文件，如下图所示修改 platform.ini 文件中的内容，重启开发环境，强制使用 Ver3.5.0 的 Espressif32 框架。



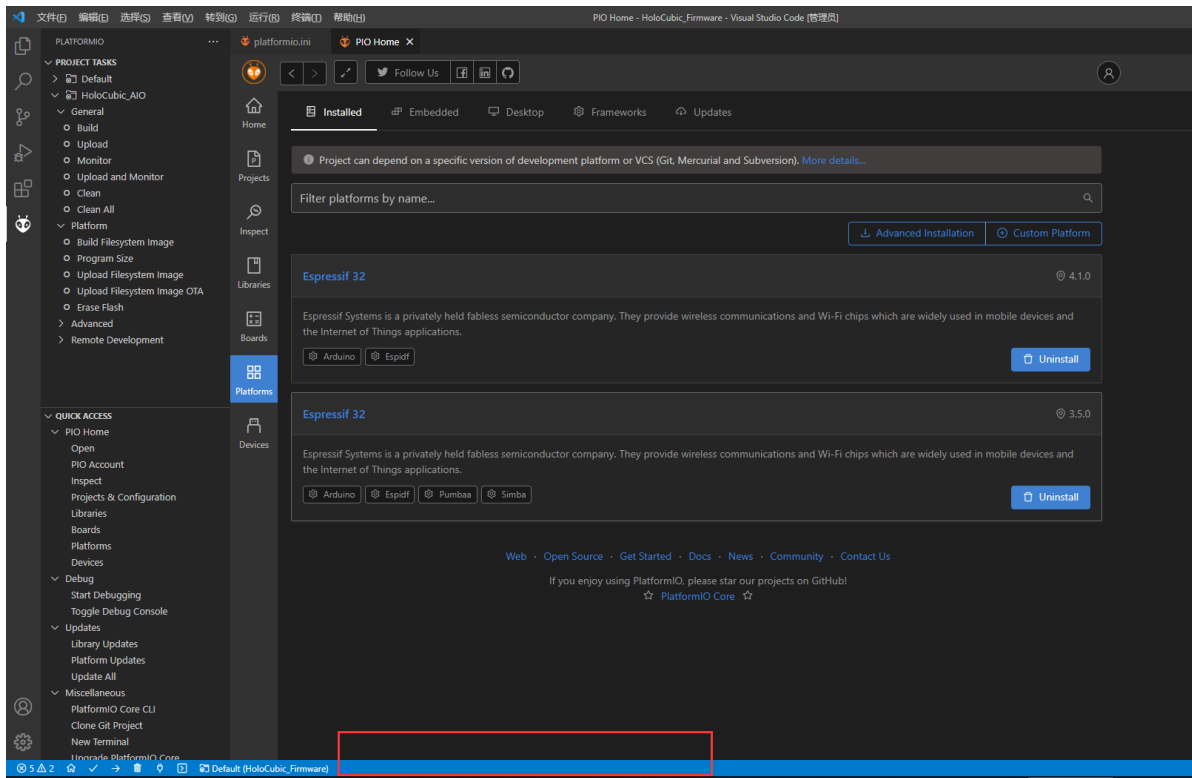
```
platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:HoloCubic_AIO]
12 ;platform = espressif32
13 platform = espressif32 @ ~3.5.0
14 ; board = esp32dev
15 board = pico32
16 framework = arduino
17
18 monitor_filters = esp32_exception_decoder
19 monitor_speed = 115200
20 ; monitor_flags =
21 ; --eol=CRLF
22 ; --echo
23 ; --filter=esp32_exception_decoder
24
25 upload_port = COM3
26 ; upload_port = COM6
27 upload_speed = 921600
28 board_build.partitions = partitions-no-ota.csv
29 board_build.f_cpu = 240000000L
30 board_build.f_flash = 80000000L
31 board_build.flash_mode = qio
```

修改ini文件中，这里强制使用3.5的esp32架构

修改为自己的小电视对应的COM

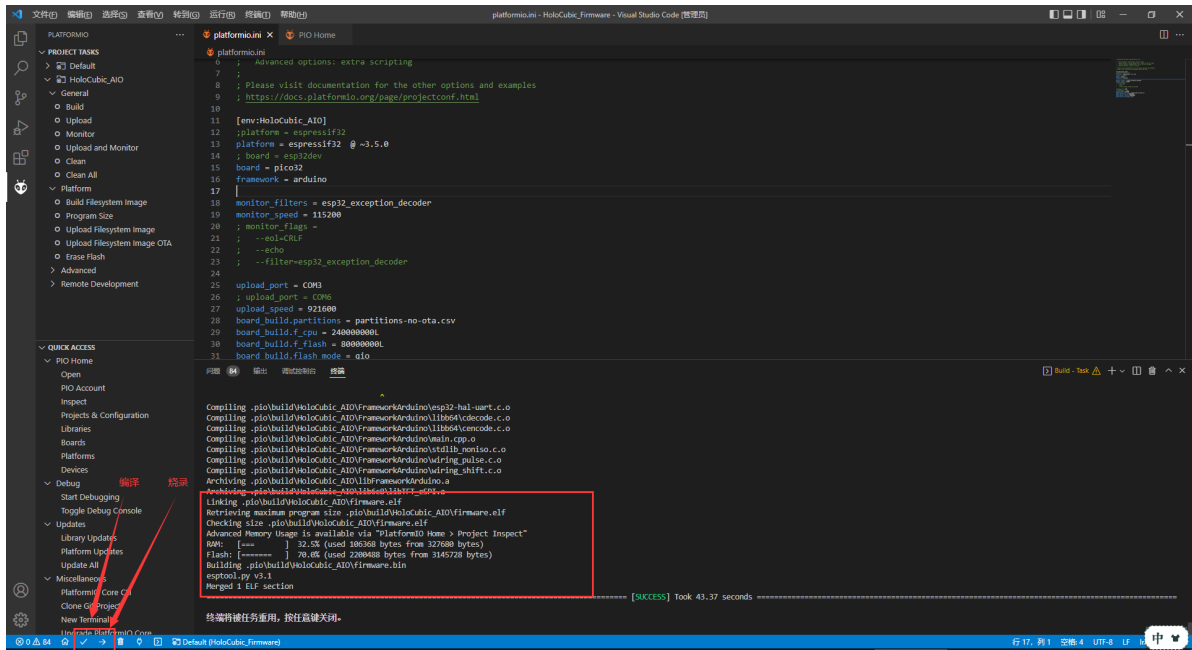
4) Rebuild

重启后，重新打开 AIO 工程，下图所示红框内会有 PlatformIO: Loading tasks... 字样，十几秒钟后，字样消失，即可进行编译和烧录。



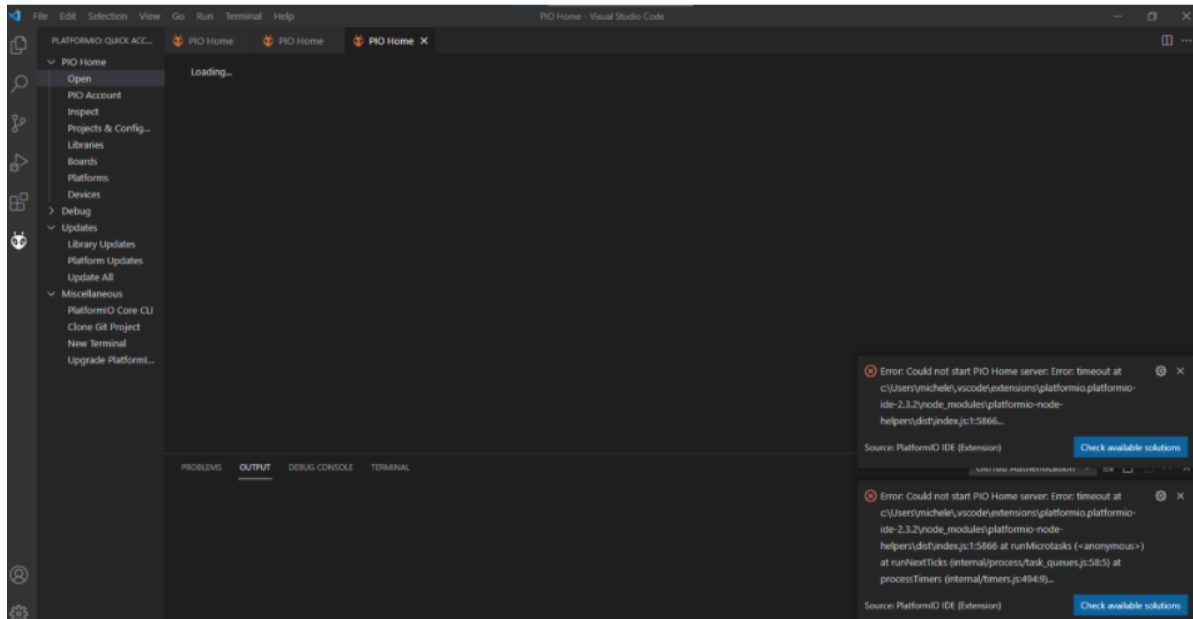
5) 编译和烧录

上一步的字样消失后，即可点击下图所示按钮进行编译和烧录。编译成功后就如下图所示，显示绿色的“success”字样。



4.FAQ

Q1: PIO主页加载失败, 如下图所示, 卡在loading。



A1: 按照下图所示步骤删除setting内的文件, 保存



重启VS Code, 重新点击蚂蚁头, 即可加载成功, 如图所示:

