

在 Developer Kit 上完成 AliOS Things 第一个应用：Hello World

摘要： 这样理解一下： VSCode 编译代码后，调用 ST-Link 将程序烧写到单片机内部。编译完成后，也会生成二进制文件，这样就可以通过各种方法下载了，比如批量生产时候用脱机下载器。

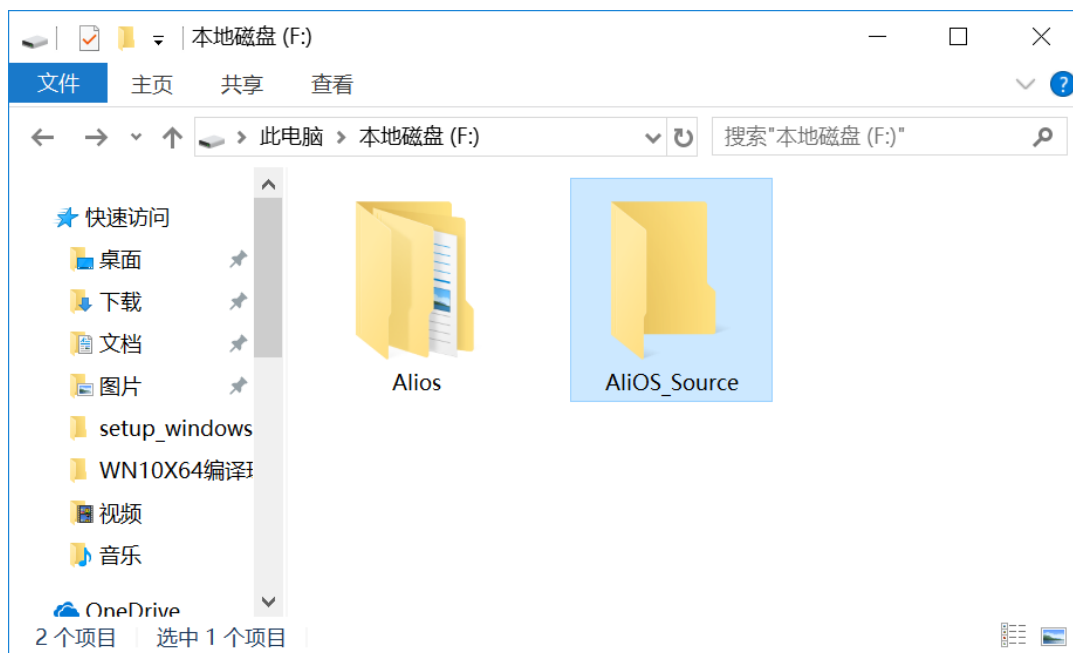
全局掌控

- 从 Git 上下载源码
- 用 VSCode 打开源码，查看源码的目录结构
- 打开 HelloWorld
- 确认手中的硬件。零妖的硬件型号是 Developer Kit 。
- 编译源码
- 确认硬件，烧录代码。打开串口查看程序输出的信息。

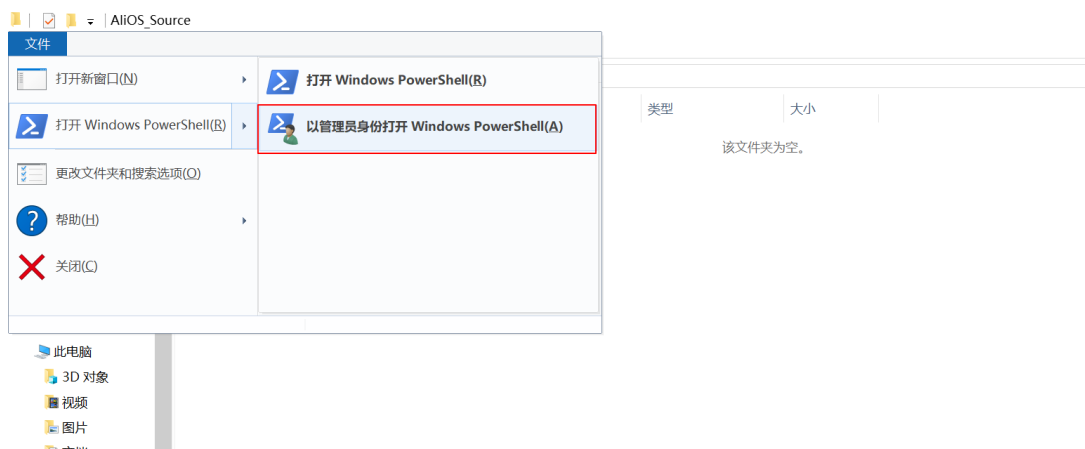
从 Git 上下载源码

新建一个文件夹，用来存放源码。

比如我在电脑的 F 盘根目录下，新建一个 AliOS_Source 的文件夹。

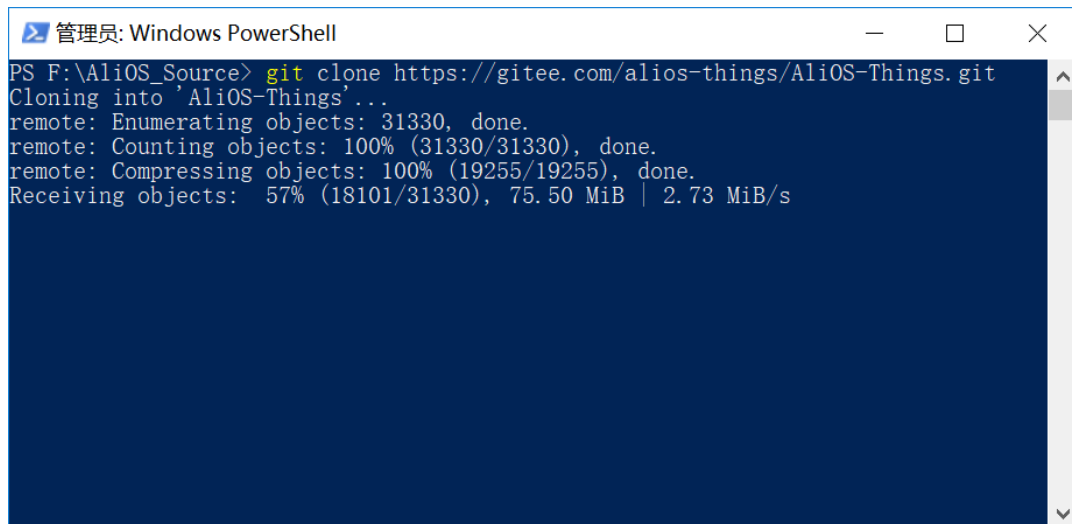


打开这个文件夹，然后打开命令行。如果你是 WIN7 等系统，可以打开 CMD 命令行，进入这个文件夹。



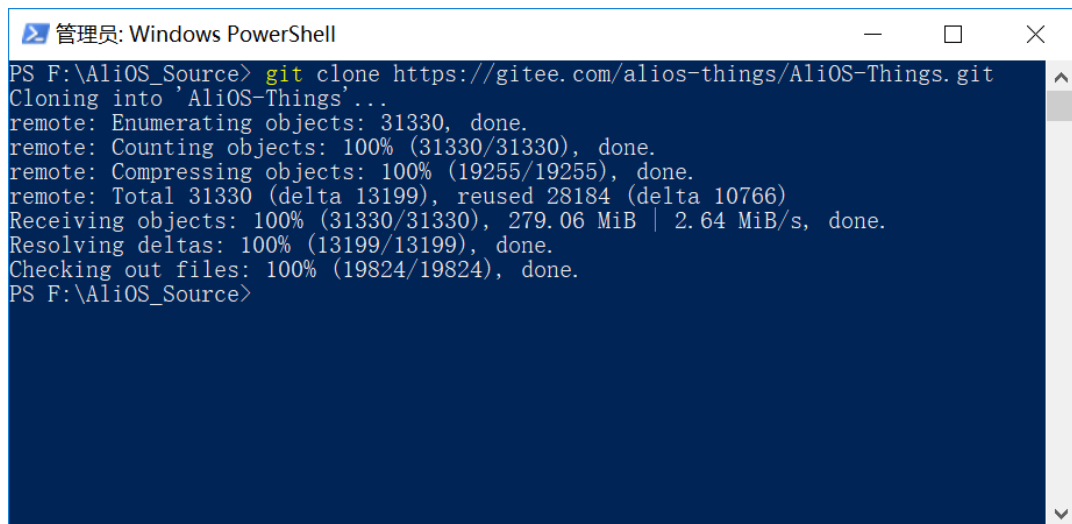
输入指令 `git clone https://gitee.com/alios-things/AliOS-Things.git` 敲回车。

文件比较大，下载时间看你的网速。



```
管理员: Windows PowerShell
PS F:\AliOS_Source> git clone https://gitee.com/alios-things/AliOS-Things.git
Cloning into 'AliOS-Things'...
remote: Enumerating objects: 31330, done.
remote: Counting objects: 100% (31330/31330), done.
remote: Compressing objects: 100% (19255/19255), done.
Receiving objects: 57% (18101/31330), 75.50 MiB | 2.73 MiB/s
```

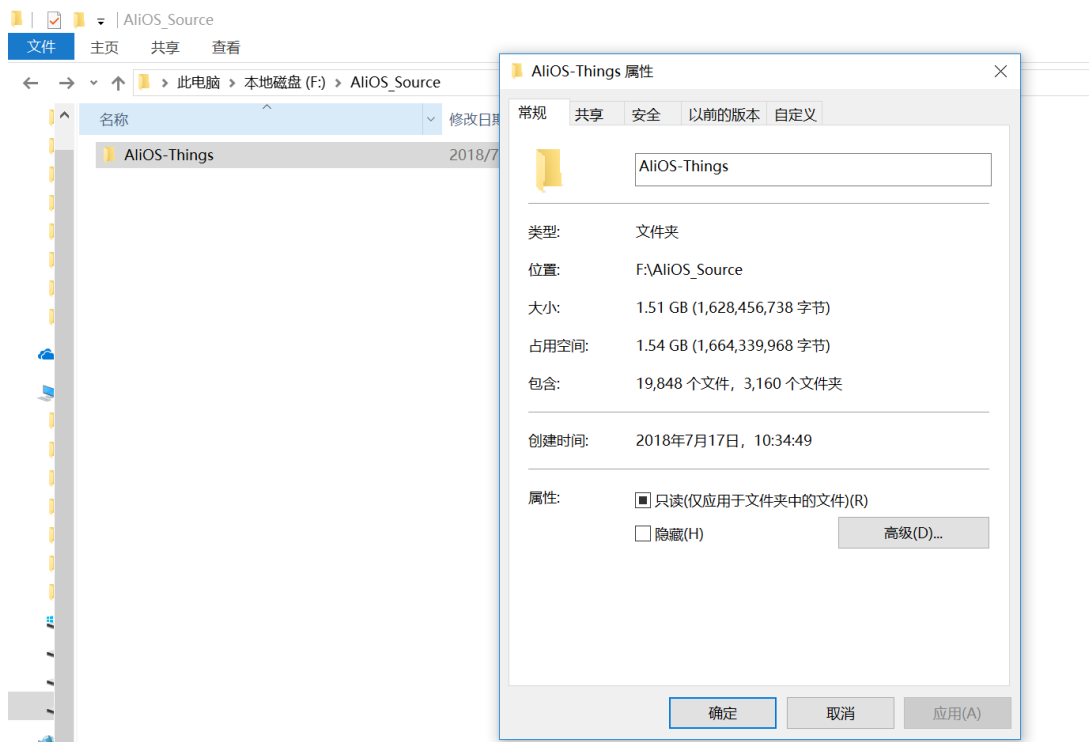
花了大概 3 分钟左右，下载完成。



```
管理员: Windows PowerShell
PS F:\AliOS_Source> git clone https://gitee.com/alios-things/AliOS-Things.git
Cloning into 'AliOS-Things'...
remote: Enumerating objects: 31330, done.
remote: Counting objects: 100% (31330/31330), done.
remote: Compressing objects: 100% (19255/19255), done.
remote: Total 31330 (delta 13199), reused 28184 (delta 10766)
Receiving objects: 100% (31330/31330), 279.06 MiB | 2.64 MiB/s, done.
Resolving deltas: 100% (13199/13199), done.
Checking out files: 100% (19824/19824), done.
PS F:\AliOS_Source>
```

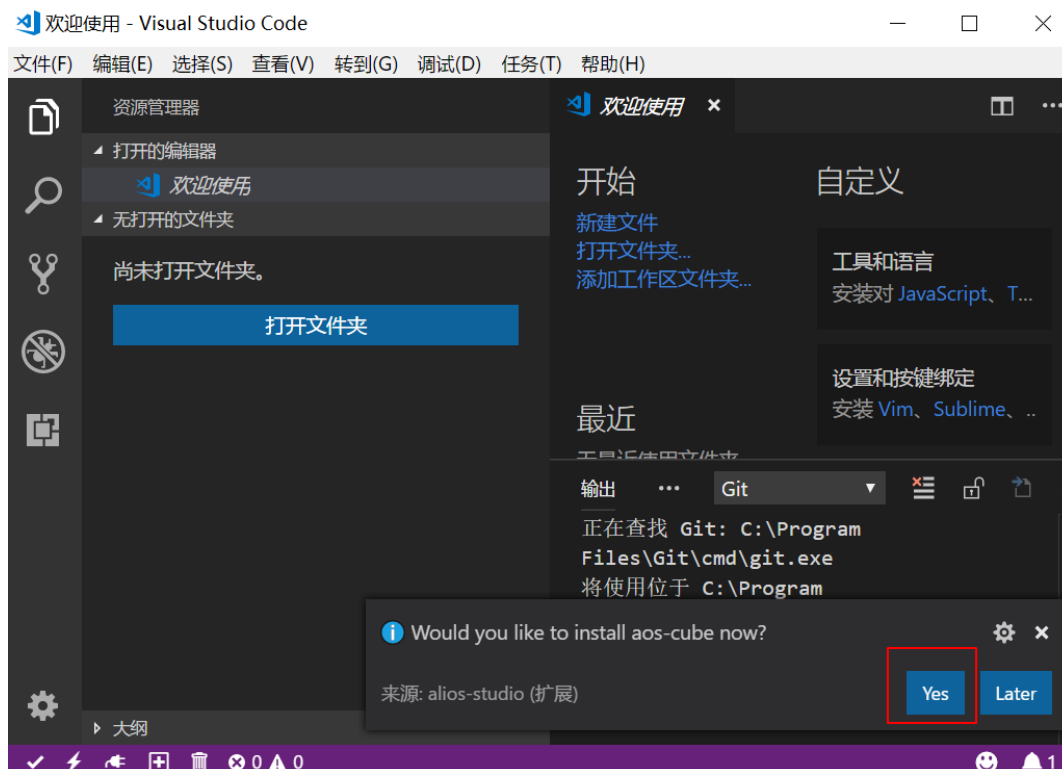
目录下多了一个文件夹，看截图，大小 1.5G。

里面就是刚刚下载的源码。



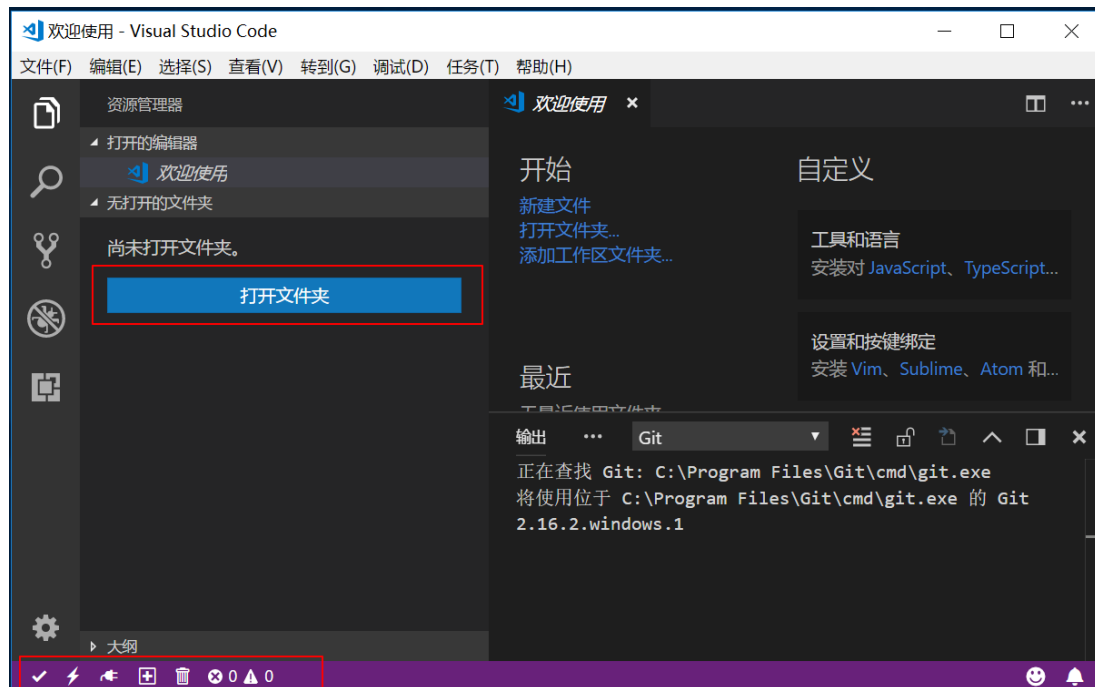
用 VSCode 打开源码文件夹，查看目录结构

打开 VSCode，再次自动弹出提示，点 Yes。



打开文件夹。就是刚才下载的那个源码文件夹。

左下角那部分，就是 AliOS 的插件。



打开文件夹后，左边显示的是文件夹里面的目录结构，可以点开，再打开文件。

每个目录里面的内容是啥，作用是啥等等，后面文章再详细介绍。

左下角，就是非常有用的控制按钮。

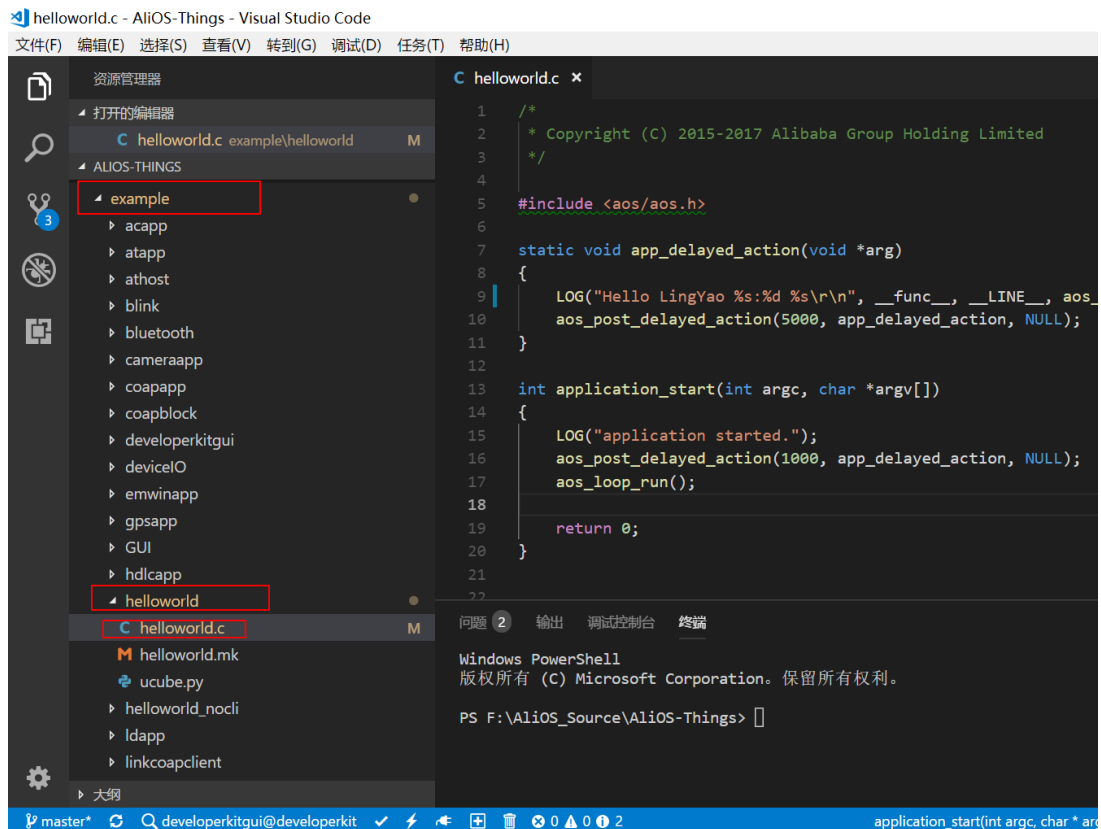


打开 HelloWorld

看截图。

在左边目录中，依次打开即可。

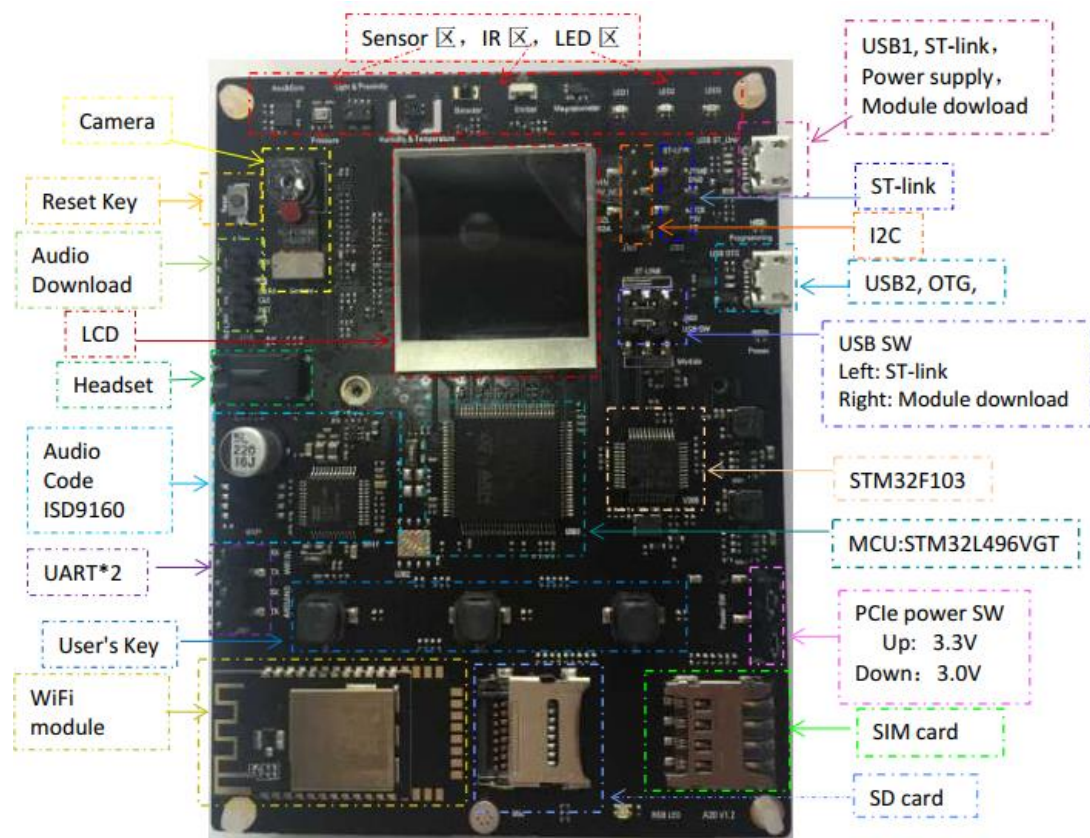
在 `example` 文件夹里面，是一个个的小示例项目。



确认手中的硬件

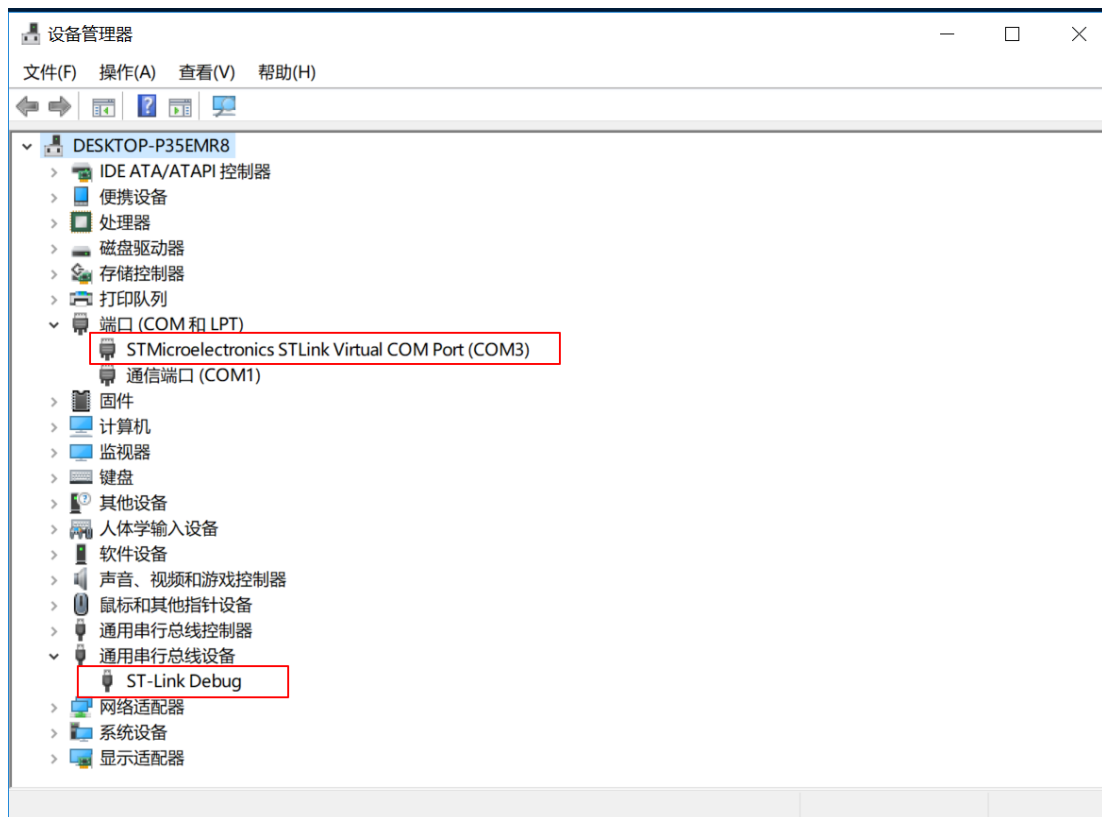
- 开发板，型号是 Developer Kit 。
- 只是作为学习测试，拿来的一块板子实验用的。
- 板子上就是一个 STM32L496 的单片机。
- 板子集成一个 ST-Link
- 板子集成一个虚拟串口
- 注意查看你自己电脑上面的串口号，零妖的是 COM3，你的需要自己查看。

此开发板淘宝有售：<https://detail.tmall.com/item.htm?spm=a230r.1.14.20.d6205eb8ofQZcH&id=571251003265&ns=1&abbucket=14>



使用数据线，插上 USB1，连接到电脑。设备管理器中会新增两个设备，分别是 ST-Link 和串口。

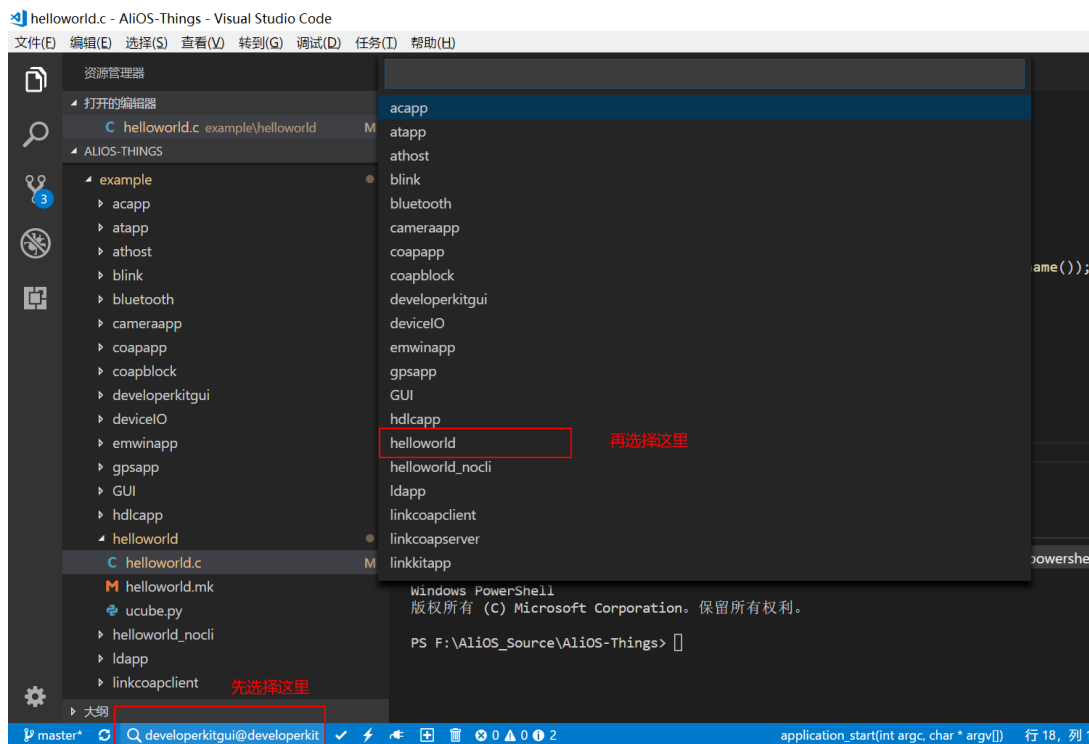
注意：新增的串口号是 **COM3** 。



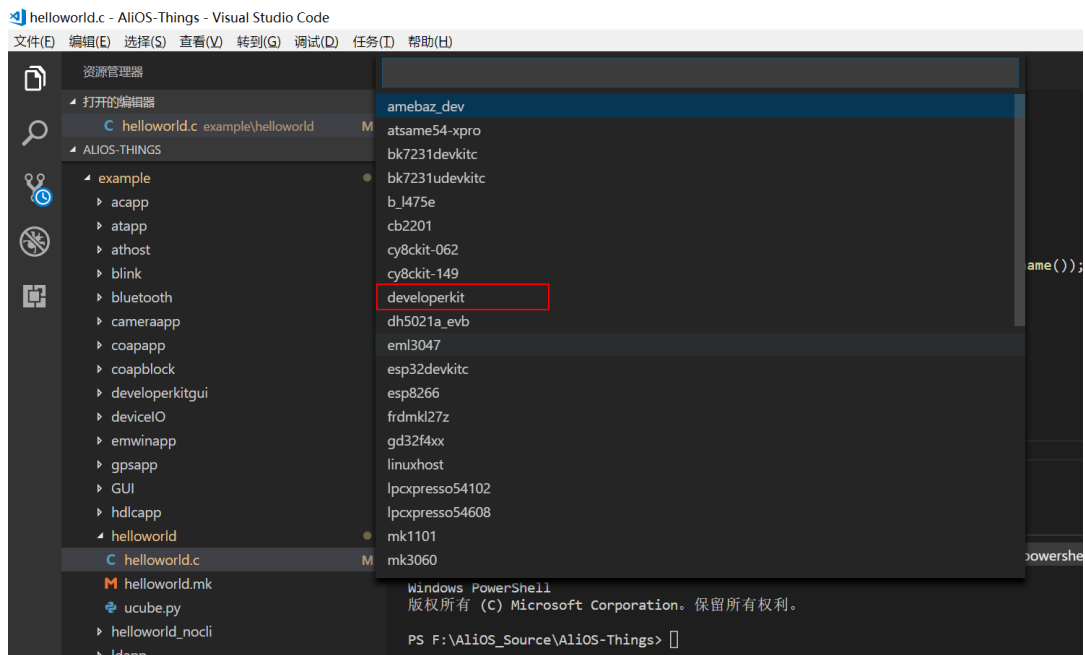
编译源码

- 在管理项目的时候，遵守 `app@board` 的原则。 `app` 表示项目的名称， `board` 表示设备的型号。
- 比如 `starterkitgui@starterkit` 表示：有一个设备型号是 `starterkit` ，项目名称是 `starterkitgui`
- 比如 `helloworld@developerkit` 表示：有一个设备型号是 `developerkit` ，项目名称是 `helloworld` 。
- 在编译源码的时候，要选择你要编译的项目，再选择对应的设备型号。
- 设备的型号可以是一个开发板比如 `developerkit`，也可以是一个芯片的型号比如 `gd32f4xx`，还可以是一个模组的型号比如 `mk3060` 等。

点击左下角后，自动弹出一个列表。选择要编译的项目名称。



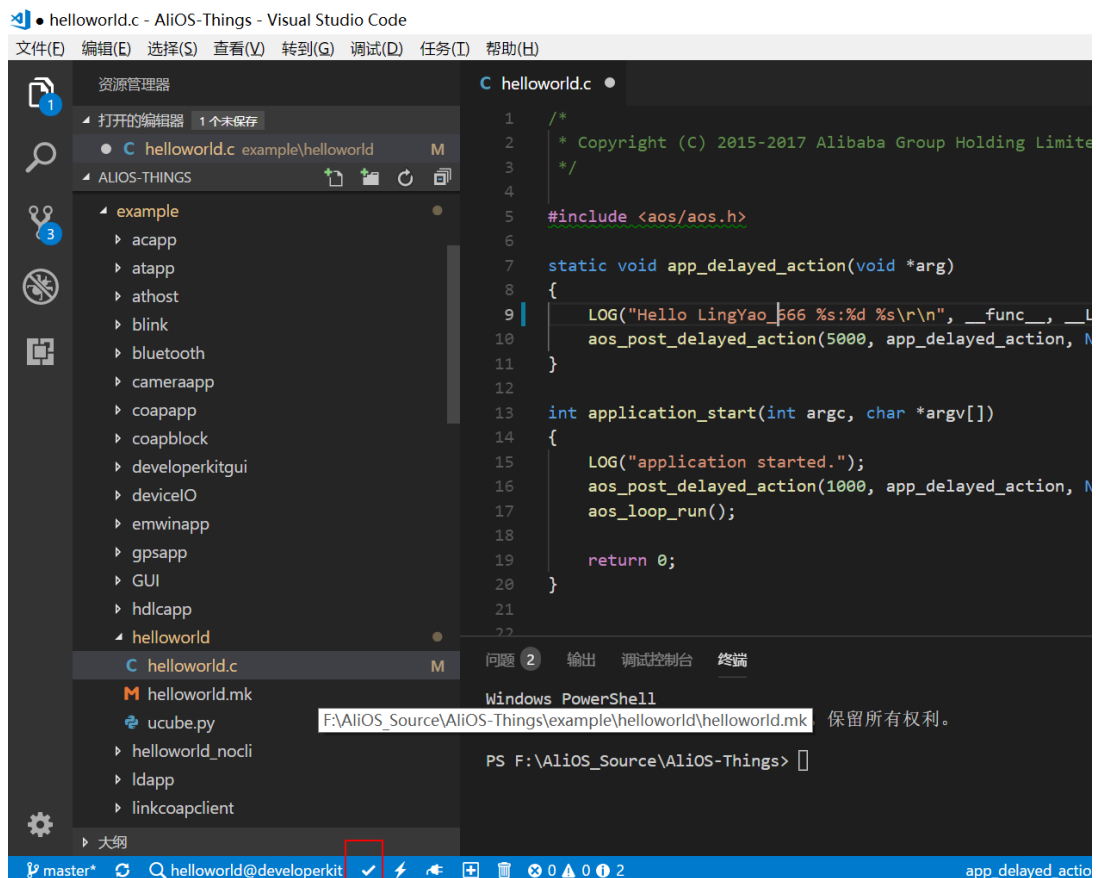
自动弹出第二轮列表，选择对应的设备型号。



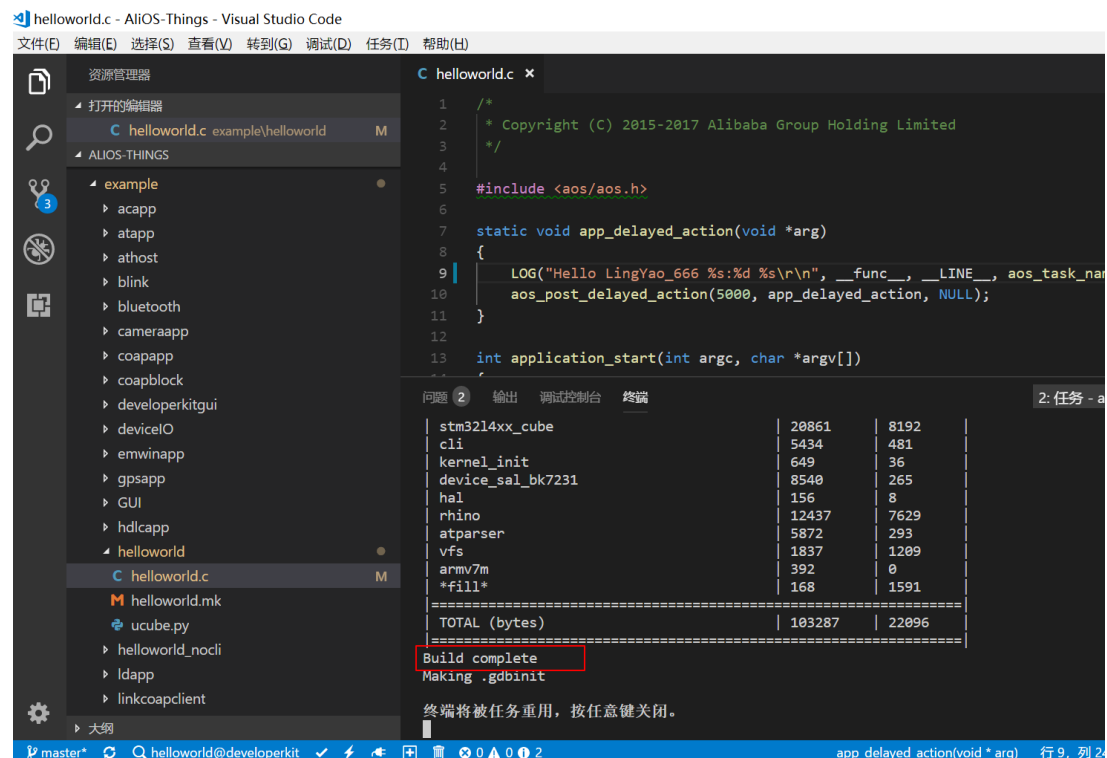
简单修改一下源码，你可以修改一下，随意输出一个字符串。比如改成你的姓名。

```
C helloworld.c •
1  /*
2   * Copyright (C) 2015-2017 Alibaba Group Holding Limited
3   */
4
5  #include <aos/aos.h>
6
7  static void app_delayed_action(void *arg)
8  {
9  |  LOG("Hello LingYao_666 %s:%d %s\r\n", __func__, __LINE__, aos_task_name());
10     aos_post_delayed_action(5000, app_delayed_action, NULL);
11  }
12
13  int application_start(int argc, char *argv[])
14  {
15     LOG("application started.");
16     aos_post_delayed_action(1000, app_delayed_action, NULL);
17     aos_loop_run();
18
19     return 0;
20 }
21
```

保存一下。然后点击左下角的 Build 按钮。编译项目。

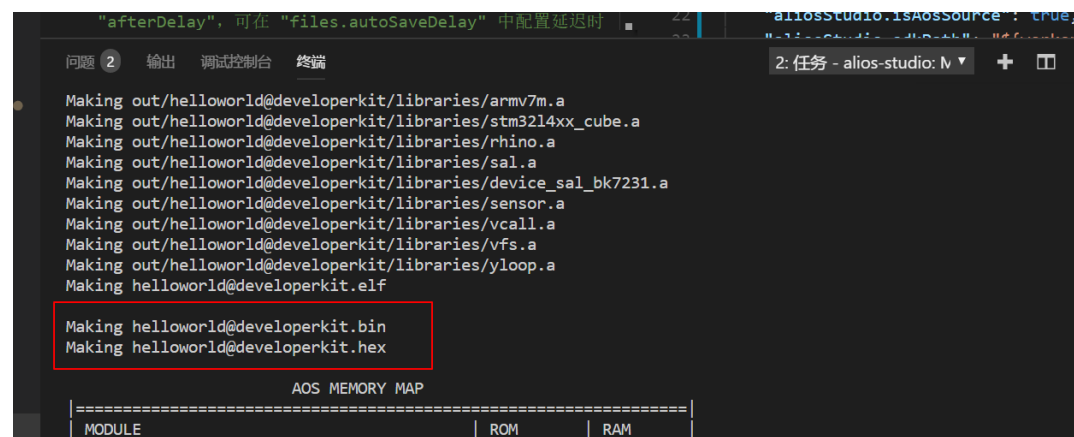


编译完成后，提示了一个成功信息。



编译成功后，还同时生成了二进制文件

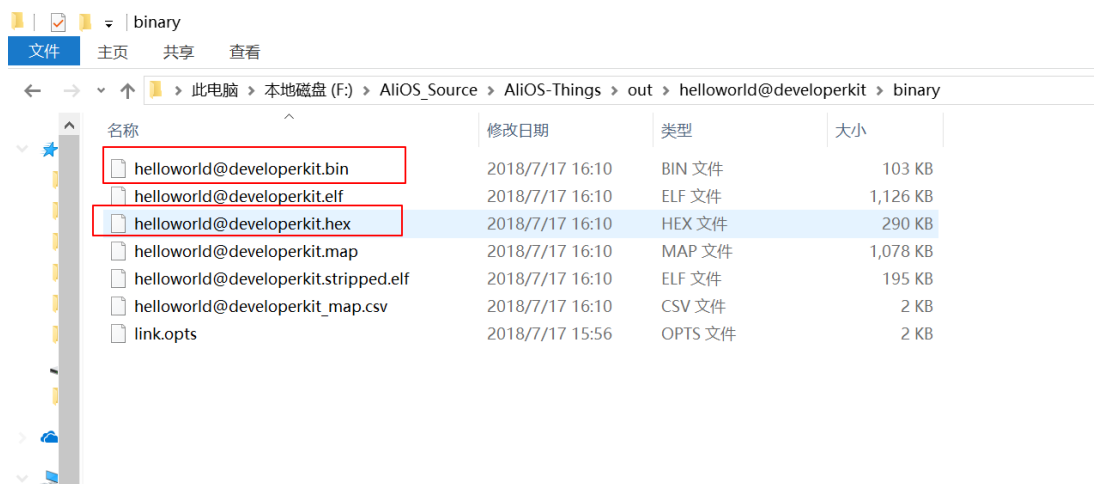
方便你用其他方法烧写到单片机，比如工厂批量用的脱机下载器



二进制文件的路径是 `\AliOS-Thingsouthelloworld@developerkitbinary`

比如零妖的路径：

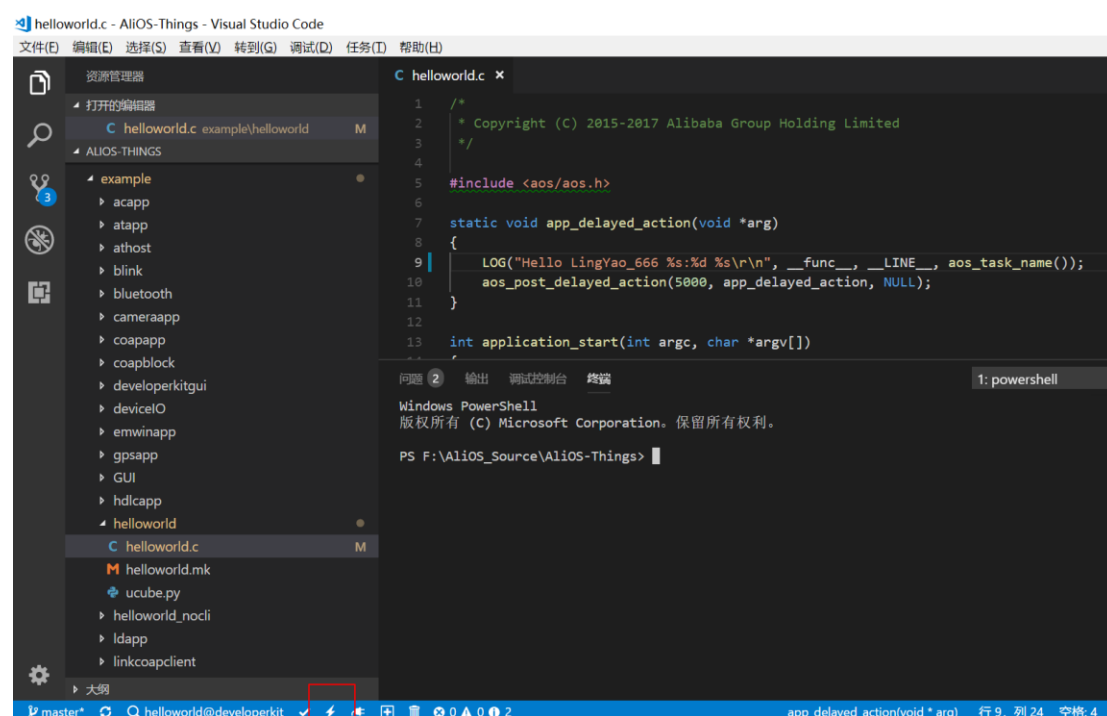
`F:\AliOS_Source\AliOS-Thingsouthelloworld@developerkitbinary`



烧录代码，打开串口，查看程序输出

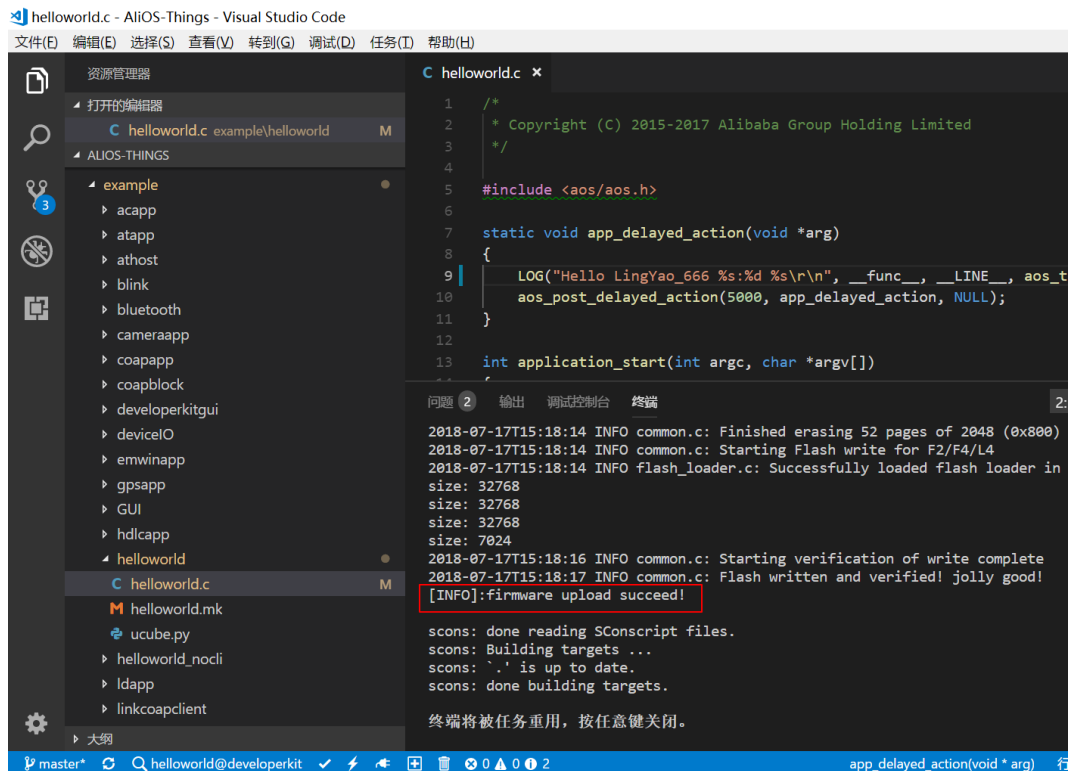
点击 Upload 按钮。

程序会调用 ST-Link，将代码烧写到单片机里面。



这个提示信息表示：

烧写成功！



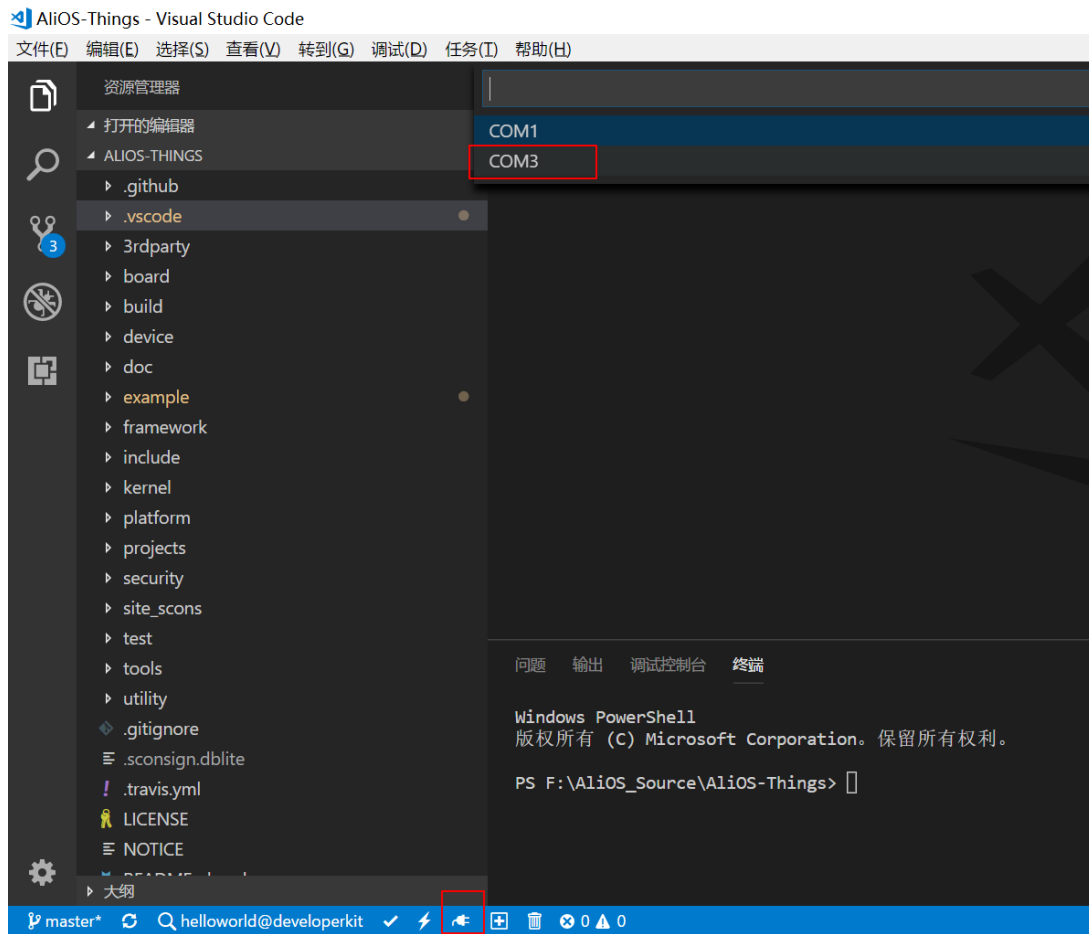
- 按任意键，会退出刚才的窗口。
- 程序的功能是通过串口定时输出一个字符串。
- 你需要打开一个串口助手查看。
- 你也可以直接在 VSCode 里打开串口查看。
- 在零妖的电脑上，打开的是 COM3 ， 波特率 115200 。

教你在 VSCode 里面打开串口

- 零妖的设备型号是 Developer Kit，只要插上电脑就会有一个串口。
- 如果你是 USB 转 TTL，你需要插上。
- 点 Connect Device 按钮。

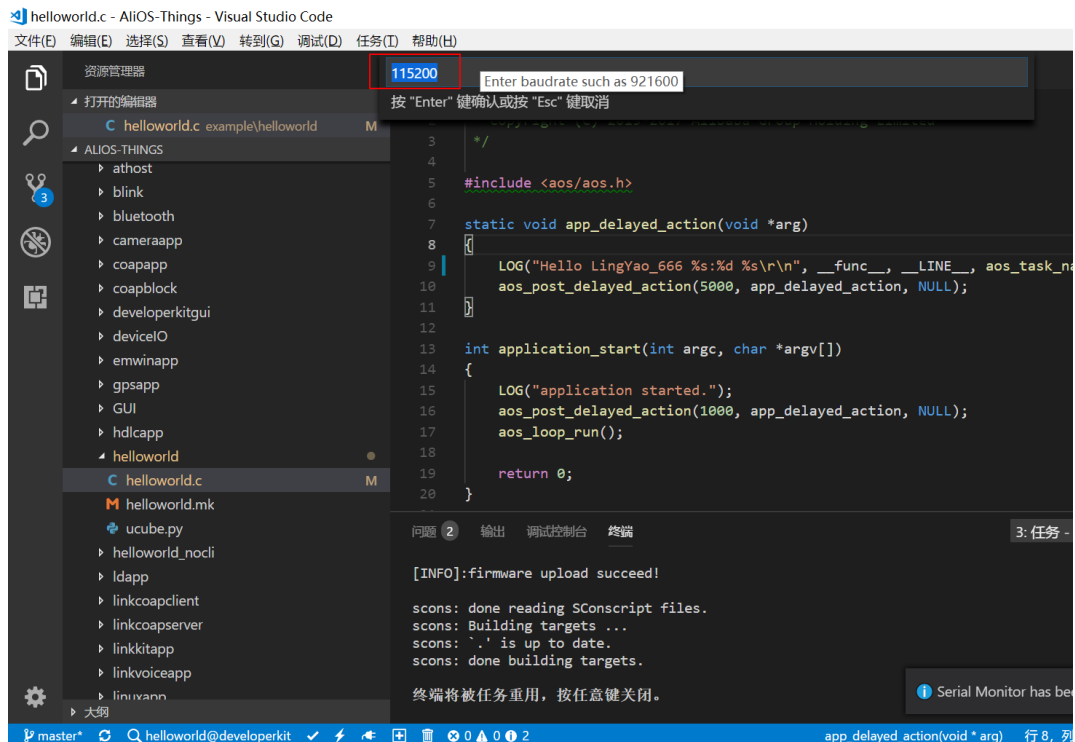
如果是第一次点它：

选择对应的 COM 口。

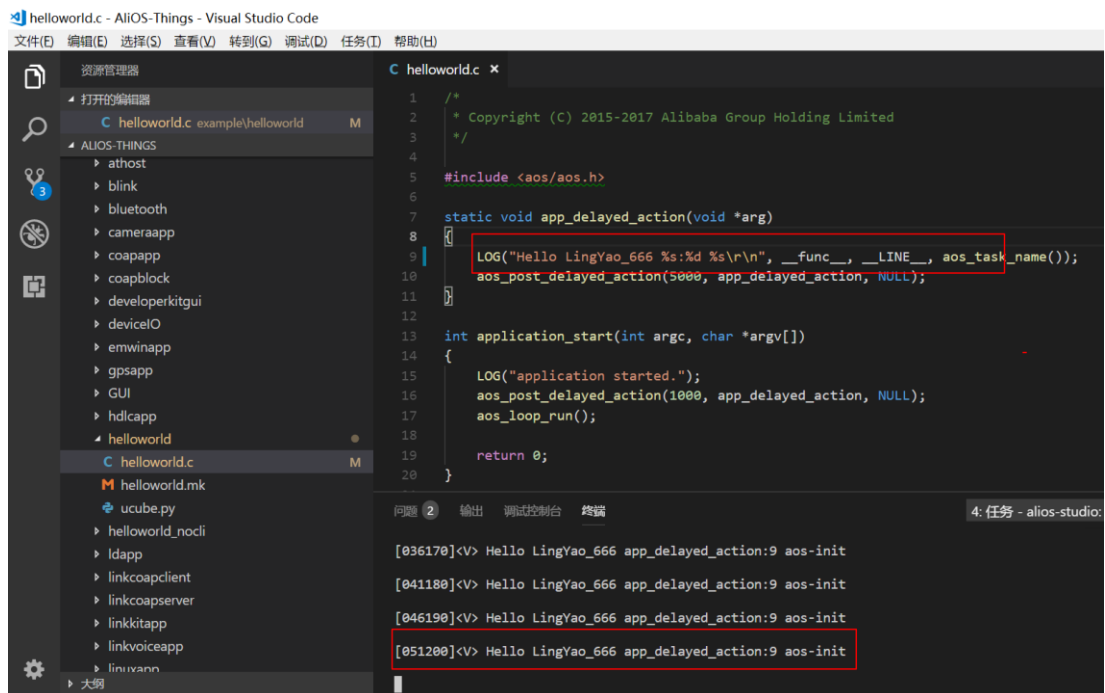


输入对应的波特率。AliOS 的示例代码中，给单片机初始化的波特率是 115200

。



查看单片机给电脑发送的信息，和写的代码一致。



如果第一次选错了，想修改：



完成

恭喜你，已经入门了 AliOS Things ！

这样理解一下： VSCode 编译代码后，调用 ST-Link 将程序烧写到单片机内部。

编译完成后，也会生成二进制文件，这样就可以通过各种方法下载了，比如批量生产时候用脱机下载器。

感谢开发者朋友们的分享，原文链接：

<https://yq.aliyun.com/articles/613760?spm=a2c4e.11155515.0.0.t55TdK>