

Professional development environment for



PlatformIO IDE(VSCode) 基本使用 - 新建项目



十里

中国石油大学（华东）控制工程硕士

34 人赞同了该文章

PlatformIO IDE (VSCode) 可以帮助我们更好地使用 PlatformIO，一个 MCU 项目的开始那就是新建，本文就讲解一下如果使用 PlatformIO IDE 新建一个 MCU 的项目。

1. 基本概念

在使用 PlatformIO 的过程中经常会遇到一些词，比如 Platform、Frameworks 以及 Boards，在新建项目之前有必要先说明一下，这些具体都代表了什么！

1.1 Platform

直译的话就是 **平台**，具体就是指的芯片平台，再详细一点那就是各个公司具体的系列芯片的开发平台了。目前为止 PIO^[1] 针对支持的平台都有以下功能支撑：

- 支持指定框架的基于脚本的编译构建系统
- 针对各公司常规开发板的预配置
- 提供多架构的构建工具及相关工具链的支持

PIO 目前支持的平台分为嵌入式和桌面两大类。

- 嵌入式平台
 - [Aceinna IMU](#)
 - [Atmel AVR](#)
 - [Atmel SAM](#)
 - [Espressif 32](#)
 - [Espressif 8266](#)
 - [Freescale Kinetis](#)
 - [Infineon XMC](#)
 - [Intel ARC32](#)



- [ST STM32](#)
 - [ST STM8](#)
 - [Teensy](#)
 - [TI MSP430](#)
 - [TI TIVA](#)
 - [WIZNet W7500](#)
-
- 桌面平台
 - [Native](#)
 - [Linux ARM](#)
 - [Linux i686](#)
 - [Linux x86_64](#)
 - [Windows x86](#)

上面也有前面文章中使用到的 ST STM32, Atmel AVR 一般指的就是 Arduino 系列开发板使用的芯片, 说到 Arduino, 它其实就是一个完整的生态, 提供开源的硬件开发板的设计以及 Arduino 统一的软件开发包就是所谓的 Frameworks。

1.2 Frameworks

上面已经提到 Frameworks 其实就是类似于 SDK 的一个东西, 全世界最有名的 Arduino 有自家的一套 SDK, 那就是 Arduino, 框架基本特征就是提供一整套一致的 API 的集合, 本质是一个官方或第三方提供的软件库。

因为 Arduino 极具影响力, 所以很多芯片平台也都有了自己的 Arduino 框架, 现在 Arduino 框架平台支持的平台有很多: Atmel AVR、Atmel SAM、Espressif 32、Espressif 8266、Infineon XMC、Intel ARC32、Kendryte K210、Microchip PIC32、Nordic nRF51、Nordic nRF52、T STM32、ST STM8、Teensy、TI MSP430、TI TIVA ^[2]。

反过来讲, 一个平台有可能支持多个不同开发框架, 比如 **ST STM32** 支持的开发框架有: Arduino、CMSIS、libOpenCM3、mbed、SPL、STM32Cube^[3]。

PIO 支持的框架也是有很多, 总有适合你的:

- Arduino
- ARTIK SDK
- CMSIS
- ESP8266 Non-OS SDK
- ESP8266 RTOS SDK
- ESP-IDF
- Freedom E SDK
- Kendryte FreeRTOS SDK
- Kendryte Standalone SDK
- libOpenCM3
- mbed
- PULP OS
- Pumbaa

的支持列表：

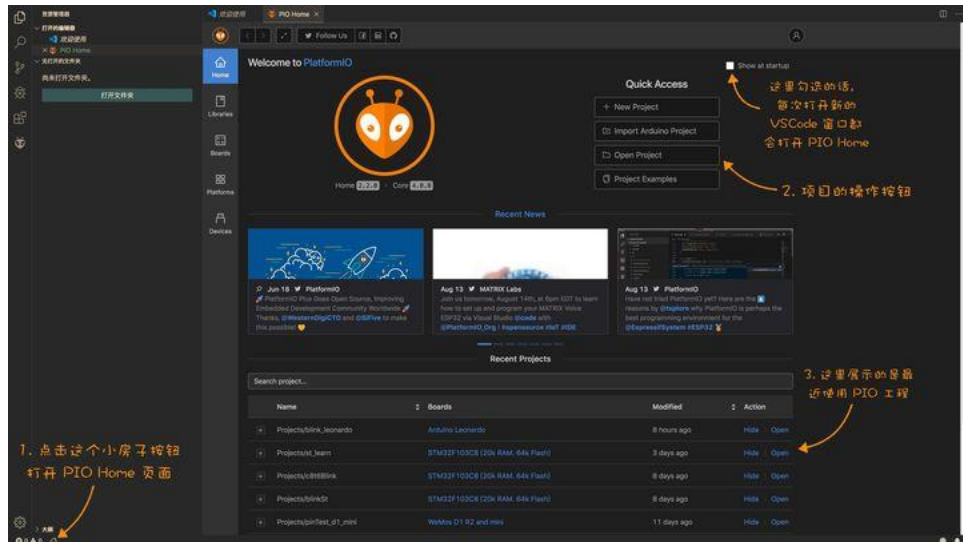
docs.platformio.org/en/...

PIO 针对支持的开发板都提供完整的项目生成脚本，新建项目从未如此快捷和简单！

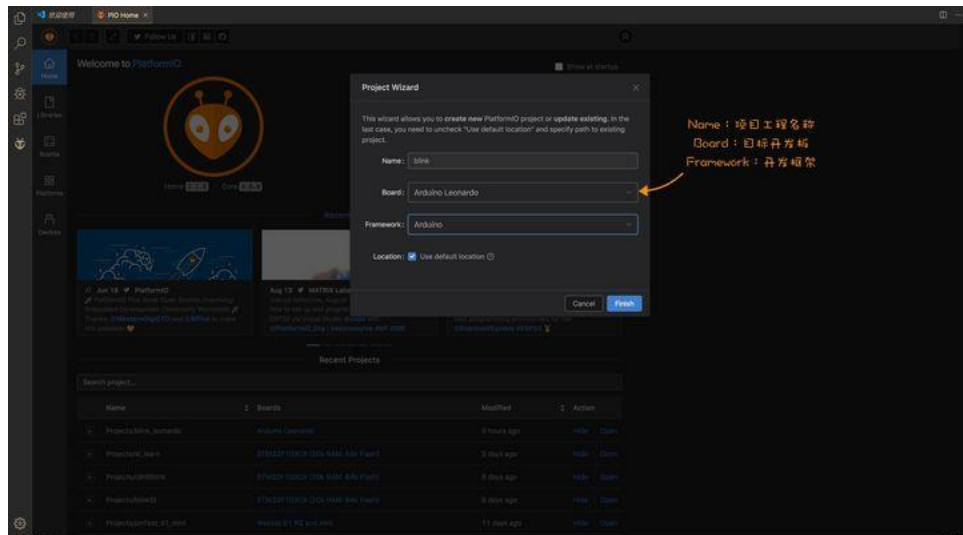
2. 新建项目

这里我们以新建一个 **Arduino Leonardo** 的项目为例。

- 打开 PIO Home：打开新的 VSCode 的窗口，点击左下角状态栏的第三个小房子按钮，就能打开 PIO Home 页面了：



- 点击项目操作按钮的 + **New Project**，在弹窗中分别填入或选择 Name、Board、Framework 点击 Finish：





3. 项目文件结构

项目文件结构要了解一下，方便后期自己手动管理项目。

- **.pio**，存放工程编译产生的文件
- **.vscode**，存放针对工程定制化的 vscode 配置文件
- **include**，存放统一管理的 h 头文件
- **lib**，存放自己编写的库文件
- **src**，存放工程项目的 C/C++ 源文件
- **test**，存放工程项目的测试文件，一般用不到
- **.gitignore**，git 仓库的忽略文件，方便 git 进行工程项目的版本控制
- **travis.yml**，持续集成的配置文件，一般用不到
- **platformio.ini**，项目的核心配置文件，这个会经常用到，所以得了解其中可用的配置项^[4]

4. 编译和上传

新建了项目工程之后，就是进行开发了，必然要用到编译代码，要将编译生成的二进制文件上传到开发板的芯片中。

为了看到效果，我们编写一段让板载 LED 闪动的实现：

```
#include <Arduino.h>

void setup() {
    // put your setup code here, to run once:
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(LED_BUILTIN, HIGH);
    delay(1000);
    digitalWrite(LED_BUILTIN, LOW);
    delay(1000);
}
```

在这之前有必要了解一下 PIO 的工具栏按钮：



Tip:

如果使用 Arduino 框架代码中，一定要包含头文件 `Arduino.h`。

4.1 编译项目工程

将上面的代码录入 `src` 目录下的 `main.cpp` 文件中，看文件名就知道使用的是 C++ 编程语言了。多种方式可以触发编译，比如笔者经常用的两种方式：

- 点击上面截图底部中工具栏的第二个对号图标
- 按下组合键 `CTRL + ALT + B`

还有两种方式可以触发编译任务：

- 按下组合键 `CTRL + SHIFT + P` (macOS 下是 `COMMAND + SHIFT + P`)，然后输入关键字 **platformio**，选择 **Build**
- 点击窗口左边栏的蚂蚁头的图标会打开 PlatformIO 的工具栏，点击其中的 **Build**

顺利的话，触发了编译，很快就会在终端窗口提示编译完成：

```
> Executing task: platformio run <

Processing leonardo (platform: atmelavr; board: leonardo; framework: arduino)
-----
Verbose mode can be enabled via `-v, --verbose` option
CONFIGURATION: https://docs.platformio.org/page/boards/atmelavr/leonardo.html
PLATFORM: Atmel AVR 1.15.0 > Arduino Leonardo
HARDWARE: ATMEGA32U4 16MHz, 2.50KB RAM, 28KB Flash
PACKAGES: toolchain-atmelavr 1.50400.190710 (5.4.0), framework-arduinoavr 4.1.1
LDF: Library Dependency Finder -> http://bit.ly/configure-pio-ldf
LDF Modes: Finder ~ chain, Compatibility ~ soft
Found 7 compatible libraries
Scanning dependencies...
No dependencies
Checking size .pio/build/leonardo/firmware.elf
Memory Usage -> http://bit.ly/pio-memory-usage
DATA:    [=          ]  5.8% (used 149 bytes from 2560 bytes)
PROGRAM: [=          ] 14.4% (used 4130 bytes from 28672 bytes)
===== [SUCCESS] Took 0.66 seconds =====
```

4.2 上传

先把手头的 Arduino Leonardo 开发板与电脑连接，直接触发上传任务即可，甚至不用自己去选



Blink Here

总结

PIO 让新建芯片开发的项目工程变得异常简单，只需几小步就可以完成，另外本文也简单说了一下程序的编译和上传操作，同样简单方便！希望本文能够帮助想要体会 PlatformIO 魅力的朋友！

博客原文：

PlatformIO IDE(VSCode) 基本使用 - 新建项目

blog.smslit.cn/2019/08/18/platformio-ne...

参考

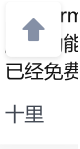
- 1. ^ PIO 是 PlatformIO 的简称，后面系列文章会经常使用这个简称
- 2. ^ 更多关于 Arduino 框架的信息参考：<https://docs.platformio.org/en/latest/frameworks/arduino.html>
- 3. ^ 更多关于 stm32 单片机开发平台的信息参考：<https://docs.platformio.org/en/latest/platforms/ststm32.html>
- 4. ^ PIO 是 PlatformIO 的简称，后面系列文章会经常使用这个简称

编辑于 07-09

Arduino MCU程序设计

文章被以下专栏收录

...



9 条评论 ⇌ 切换为时间排序

写下你的评论...

尹念龙 2020-03-16

如何更改里面的库为国内服务器下载 提高安装速度？下载太慢了

2

弓长WEI 04-23

我怎么找不到PIO的编译和下载按钮，左下角没有工具栏，有哪位大神解答一下！

赞

小瑞猫 02-25

没办法修改安装目录测试只能安装到C盘

赞

知乎用户 (作者) 回复 小瑞猫 02-25

这跟用户目录在哪个盘有关，因为pio会在用户录下创建platform目录用于存放文件，同时会在用户目录下的Documents 下创建 platformio 目录存放项目工程 😊

赞

小瑞猫 回复 知乎用户 (作者) 02-25

我的很奇怪直接丢C盘了会创建 C:/platformio 然后里面有 Python3,penv,packages,platforms这些目录,我现在在看源码看看能不能发现怎么改

赞

展开其他 1 条回复

ooopt 02-23

开局难的时候就得有这种宝典护身~ 🍀🍀

赞

知乎用户 (作者) 回复 ooopt 02-25

🍀

赞