



# Speeduino 固件设置

在开发板上设置 Speeduino 固件

## 加载 Speeduino 固件

### 概述

Speeduino 固件是驱动硬件的代码，必须在使用 ECU 之前安装到您的主板上。新固件版本会定期发布（大约每 2 个月一次），带来新功能、错误修复和性能改进，因此强烈建议您保持最新状态。

为了最大限度地简化操作，编译和安装固件的过程相当简单。大多数用户将使用 SpeedyLoader 方法安装固件

### 安装 - SpeedyLoader

将 Speeduino 固件安装到标准 Arduino Mega 2560 或 Teensy 上的最简单（也是推荐的）方法是使用 SpeedyLoader 实用程序。SpeedyLoader 负责下载固件并将其安装到 Arduino 上，而无需您自己手动编译任何代码。您可以选择已发布的最新固件，或者根据需要从较旧的固件中选择一个。SpeedyLoader 还将下载 INI 文件，并可选地下载您选择的固件的基本调音，以便将其加载到您的 TunerStudio 项目中。

- ▶ **Windows:** [32 位](#) / [64 位](#)
- ▶ **Mac:** [SpeedyLoader.dmg](#)
- ▶ **Linux:** [SpeedyLoader.ApplImage](#) （下载后需要可执行）
  - ▶ Linux 需要安装 libusb 库。例如，如果在 Debian/Ubuntu 上：

```
sudo apt-get install libusb-1.0-0 libusb-0.1-4:i386
```
- ▶ 树莓派 [SpeedyLoader.ApplImage](#)
  - ▶ Raspberry Pi / Raspbian 用户可以使用以下工具安装所需的库：

```
sudo apt-get install libusb-1.0-0 libusb-0.1-4
```

在主板上安装固件后，请参阅[连接到 TunerStudio](#)，了解有关如何配置 TunerStudio 的更多详细信息

### 安装 - 使用 Arduino IDE 手动编译



请注意，安装 Speeduino**不需要**手动编译固件，最简单（并推荐给大多数用户）的方法是使用上面描述的 SpeedyLoader。

如果您想自己编译固件或进行任何代码更改，那么发布版本和当前开发版本的源代码都是免费提供的。

#### 要求

- ▶ Windows、Mac 或 Linux PC
- ▶ [Arduino IDE](#) 。当前所需的最低版本是 1.6.7，但建议使用较新的版本。
- ▶ 最新 Speeduino 代码库的副本。见下文。
- ▶ [TunerStudio](#) 的副本，用于测试固件是否已成功上传
- ▶ 安装在 Arduino IDE 上的[时间库](#)。

#### 下载固件

获取 Speeduino 固件有两种方法：


1. 定期生成稳定的代码，并将其作为发布版本发布在 Github 上。这些代码可以在以下位置找到：[发布版本](#)
2. 如果你想要最新、最好的（有时也是最不稳定的）代码，可以克隆和更新 git 存储库。  
参见[此处](#)

#### 编译固件

- ▶ 启动 IDE，选择文件 > 打开，导航到您下载 Speeduino 的位置并打开**speeduino.ino**文件。
- ▶ 设置主板类型：[工具](#)>[主板](#)>*Arduino Mega 2560*或 Mega ADK（Teensy 和其他 Arduino 主板类型也受支持，但本指南仅解释 Arduino Mega）
- ▶ 单击左上角的验证图标（看起来像一个勾号）

此时您应该有一个已编译的固件！如果您在编译过程中遇到问题，请参阅下面的[故障排除](#)部分。



该视频介绍了从头开始在 Arduino 上安装固件的整个过程：<https://www.youtube.com/watch?v=AX9URou4JT8> 

可选（但建议）

有一个选项可用于更改编译器优化级别，这可以提高。默认情况下，IDE 使用 -Os 编译选项，该选项侧重于生成小型二进制文件。由于 Speeduino 代码的大小不是问题，但速度是一个考虑因素，因此将其更改为 -O3 会产生更好的结果（速度大约快 20%，草图大小大 40%）为此，您需要编辑 platform.txt 文件：



- 确保 Arduino IDE 没有运行
- 打开位于以下位置的 platform.txt 文件：
  - 在 Windows 上：c:\Program Files\Arduino\hardware\arduino\avr
  - 在 Mac 上： /Applications/Arduino/Contents/Resources/Java/hardware/arduino/avr/
  - 在 Linux 上：
- 在以下 3 个条目中，将 Os 更改为 O3：
  - 编译器.c.标志
  - 编译器.c.elf.flags
  - 编译器.cpp.标志
- 保存文件并重新启动 Arduino IDE

安装

一旦成功编译了固件，在主板上的安装就很简单了。

- 将 Mega 2560 插入空闲的 USB 端口
- 如果您运行的是旧版本的**Windows**，并且这是您第一次使用 Arduino，则可能需要安装 Arduino 串行芯片（USB-UART 或“USB 适配器芯片”）的驱动程序。

大多数官方主板和许多非官方版本都使用 ATmega16U2 或 8U2，而许多 Mega2560 克隆板则使用 CH340G IC。两种类型都可以正常工作。串行芯片通常可以通过外观识别：

- **ATmega16U** - 在 USB 连接器附近有一个方形 IC - 驱动程序包含在 Windows 7+、MacOS 和 Linux 中。
- **WCH CH340G** - 这款产品在 USB 连接器附近有一个矩形 IC - 使用[WCH](#)  为 Windows 提供的“CH341”驱动程序
  - 适用于其他系统（Mac、Linux、Android 等）的 WCH 原装 CH340/CH341 驱动程序可[在此处](#)  找到。
- 在 Arduino IDE 中；选择 Mega2560： *工具 > 开发板*
- 选择系统的串行端口进行上传： *工具 > 串行端口*
- 点击左上角的*上传按钮*

较旧的固件版本

如果需要，也可以通过 SpeedyLoader 安装旧固件版本

验证固件

固件现已加载到您的主板上，您现在可以继续[连接到 TunerStudio](#)。

或者，您可以使用 Arduino IDE 的串行监视器手动验证固件。这可以通过从工具菜单中选择“串行监视器”来启动。

在出现的窗口中，输入大写“S”（无引号）并按*Enter*。Mega 应该会响应安装代码版本的年份和月份（xxxx.xx）：

```
1 | Speeduino 2017.03
```

**注意：** 确保波特率设置为 115200

您还可以输入“?”以获取来自 Mega 的查询列表。

故障排除

选择了错误的 **Arduino** 开发板

如果您在尝试编译固件时看到以下（或类似）错误以及解决方案：

```
1 | scheduler.ino:317:7: error: ‘OCR4A’ was not declared in this scope
2 | scheduler.ino:323:8: error: ‘TIMSK5’ was not declared in this scope
3 | scheduler.ino:323:25: error: ‘OCIE4A’ was not declared in this scope
```



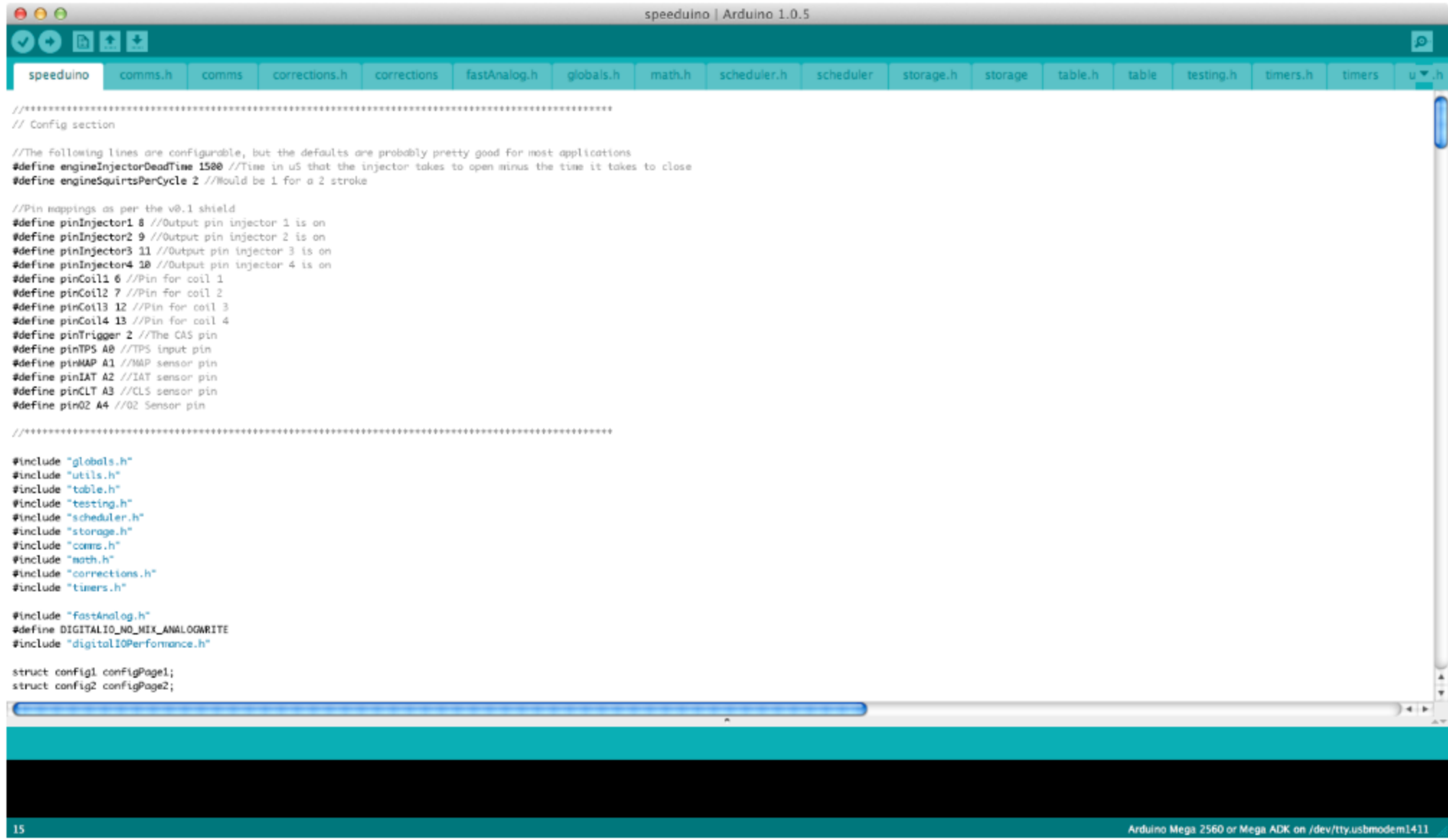
您可能选择了错误类型的 Arduino 开发板。通过选择“工具”>“开发板”>“Arduino Mega 2560或 Mega ADK”来设置开发板类型

### 整个 Speeduino 项目未打开

如果您只打开了 speeduino.ino 文件而不是整个项目，则可能会出现以下情况。

speeduino.ino:27:21: fatal error: globals.h: No such file or directory

确保所有文件都包含在同一目录中，然后选择文件->打开并找到 speeduino.ino 文件。如果您正确打开了项目，则顶部应该有多个选项卡：



如果您只看到一个文件或少数文件则表示您尚未打开整个项目。

## 安装 - 使用 Platform IO 手动编译

手动编译 Speeduino 固件的另一种方法是使用 Platform IO。使用 Platform IO 通常比使用 Arduino IDE 更容易，因为它允许在 platformio.ini 文件中自动配置项目。Platform IO 还可用于轻松构建其他 MCU 类型（Teensy、STM32）。

### 要求

- ▶ Windows、Mac 或 Linux PC
- ▶ [Visual Studio Code](#)。 [可从此处](#) 下载
- ▶ [VS Code 中的PlatformIO](#) 附加组件。要安装，请打开 VS Code 并搜索“Platform IO IDE”扩展 (Ctrl+Shift+X)，然后单击安装。
- ▶ 最新 Speeduino 代码库的副本。见下文。
- ▶ [TunerStudio](#) 的副本，用于测试固件是否已成功上传

### 下载固件

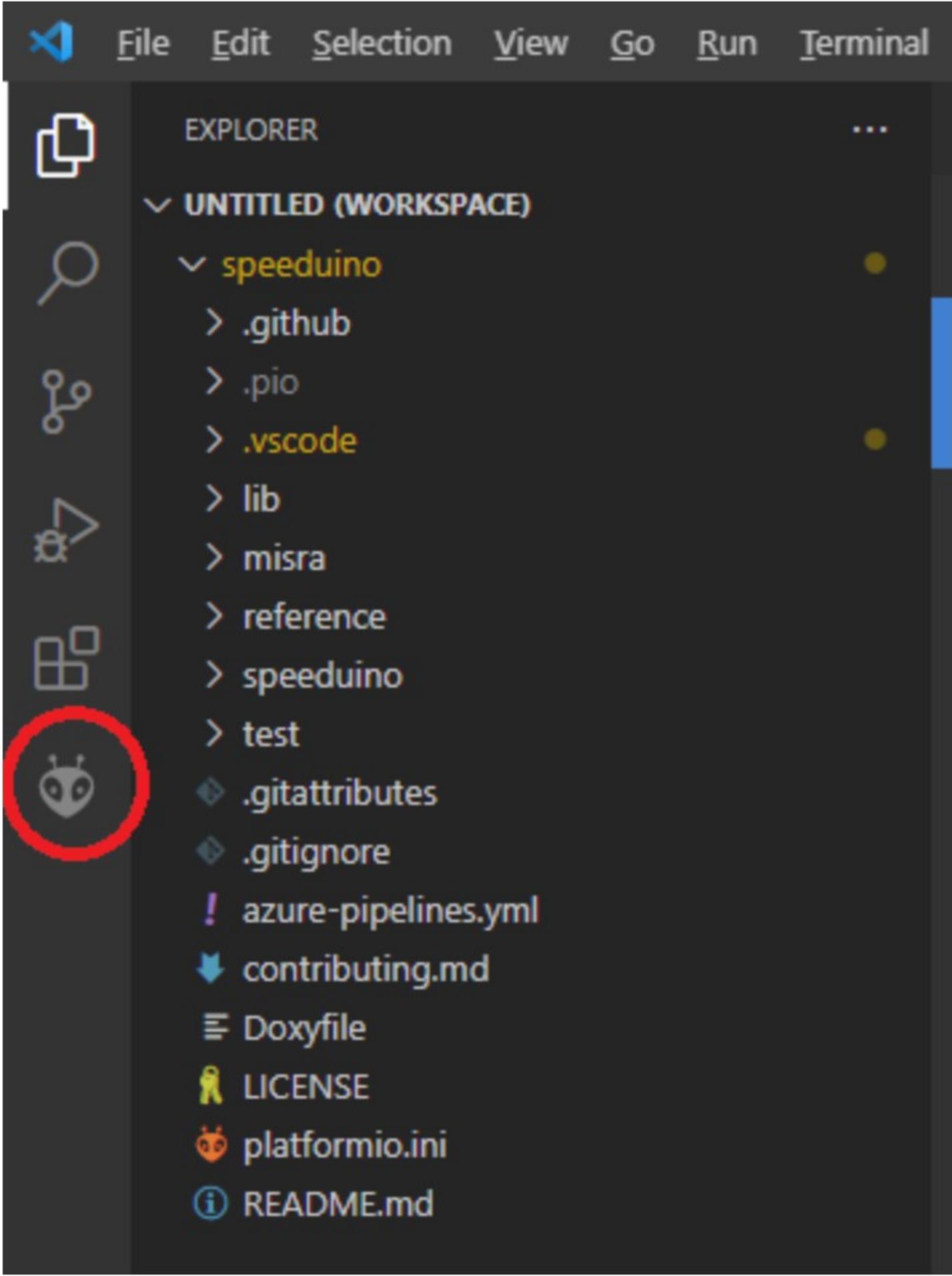
获取 Speeduino 固件有两种方法：

1. 定期生成稳定的代码，并将其作为发布版本发布在 Github 上。这些代码可以在以下位置找到： [发布版本](#)
2. 如果你想要最新、最好的（有时也是最不稳定的）代码，可以克隆和更新 git 存储库。  
参见[此处](#)

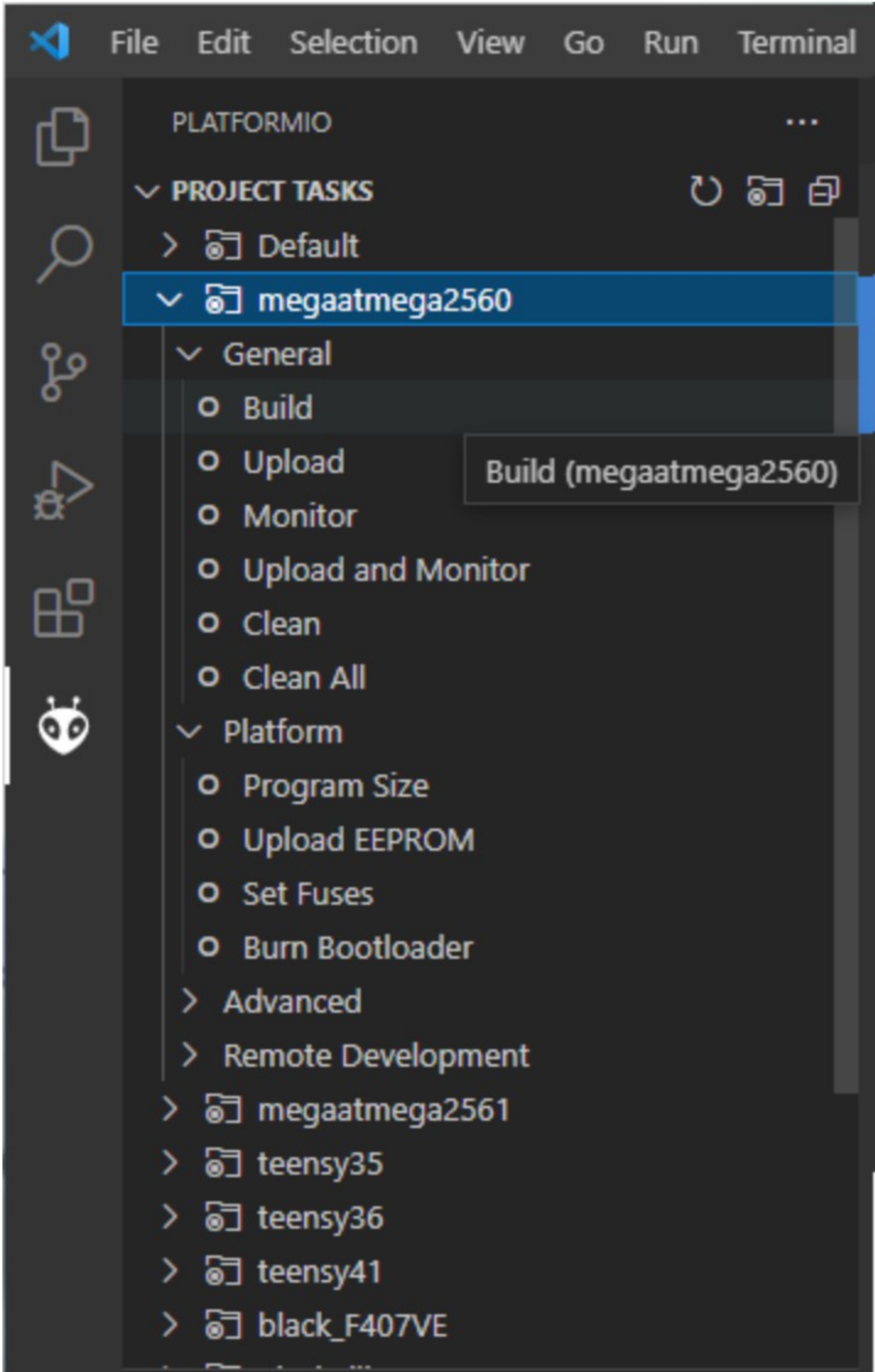
### 编译并安装固件

- ▶ 启动 VS Code 并让其打开/更新 PlatformIO 扩展，选择 *Explorer* > *打开文件夹* 并导航到下载 Speeduino 的位置
- ▶ speeduino 文件夹现在在工作区上应该如下所示（platformio.ini 可见）。单击左侧面板上的 PIO 徽标：

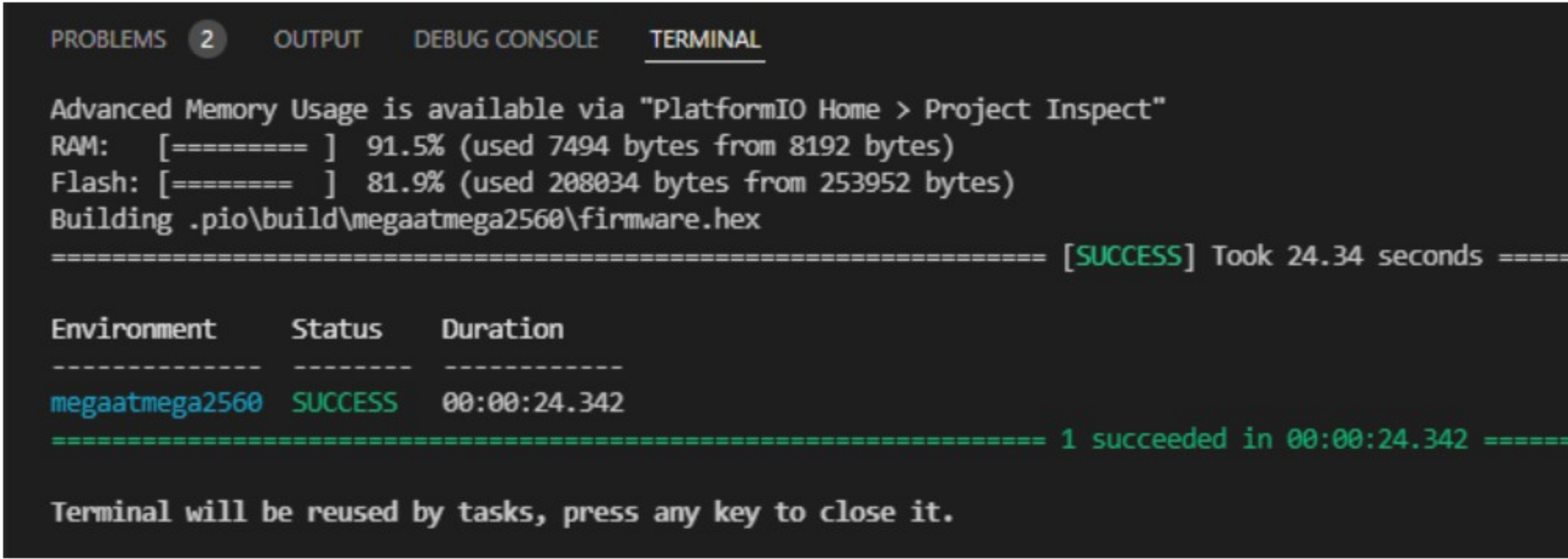




▶ 在打开的 *项目任务* 中打开 *megaatmega2560* 并单击 *Build* 来编译FW：



▶ PIO 现在应该下载编译固件所需的所有组件并进行编译。



▶ 编译完成后，您可以单击 *上传*，PIO 将把 speeduino 代码上传到 Arduino MEGA。



PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

avrdude: reading input file ".pio\build\megaatmega2560\firmware.hex"  
avrdude: writing flash (208034 bytes):  
  
Writing | ##### | 100% 31.00s  
  
avrdude: 208034 bytes of flash written  
avrdude: verifying flash memory against .pio\build\megaatmega2560\firmware.hex:  
avrdude: load data flash data from input file .pio\build\megaatmega2560\firmware.hex:  
avrdude: input file .pio\build\megaatmega2560\firmware.hex contains 208034 bytes  
avrdude: reading on-chip flash data:  
  
Reading | ##### | 100% 23.92s  
  
avrdude: verifying ...  
avrdude: 208034 bytes of flash verified  
  
avrdude: safemode: Fuses OK (E:FD, H:D8, L:FF)  
  
avrdude done. Thank you.  
  
===== [SUCCESS] Took 73.71 seconds ===  

Environment	Status	Duration
-----	-----	-----
megaatmega2560	SUCCESS	00:01:13.706

===== 1 succeeded in 00:01:13.706 =====

Terminal will be reused by tasks, press any key to close it.

通过从项目任务中选择正确的 MCU 选项，平台 IO 也可用于编译其他平台的固件。目前支持 Teensy 3.5、Teensy 3.6 和 STM32F407 在汽车上运行固件。