

硬件要求

硬件要求概述

本页介绍了 Speeduino 系统的基本硬件要求，以及一些不同版本支持的选项。它并不涵盖所有支持的硬件组合，但可以为初学者提供一些概述。

Arduino

大多数 Speeduino 设计都使用 Arduino Mega 2560 R3 作为控制器。所有官方和大多数 Arduino Mega 2560 克隆板都可以正常工作，但建议使用具有 16u2 串行接口的开发板，而不是更便宜的 CH340。开发板使用的芯片通常可以在大多数零售商的信息/规格列表中找到，但如果有疑问，请咨询您要购买的卖家。

输入

曲轴传感器/凸轮轴传感器

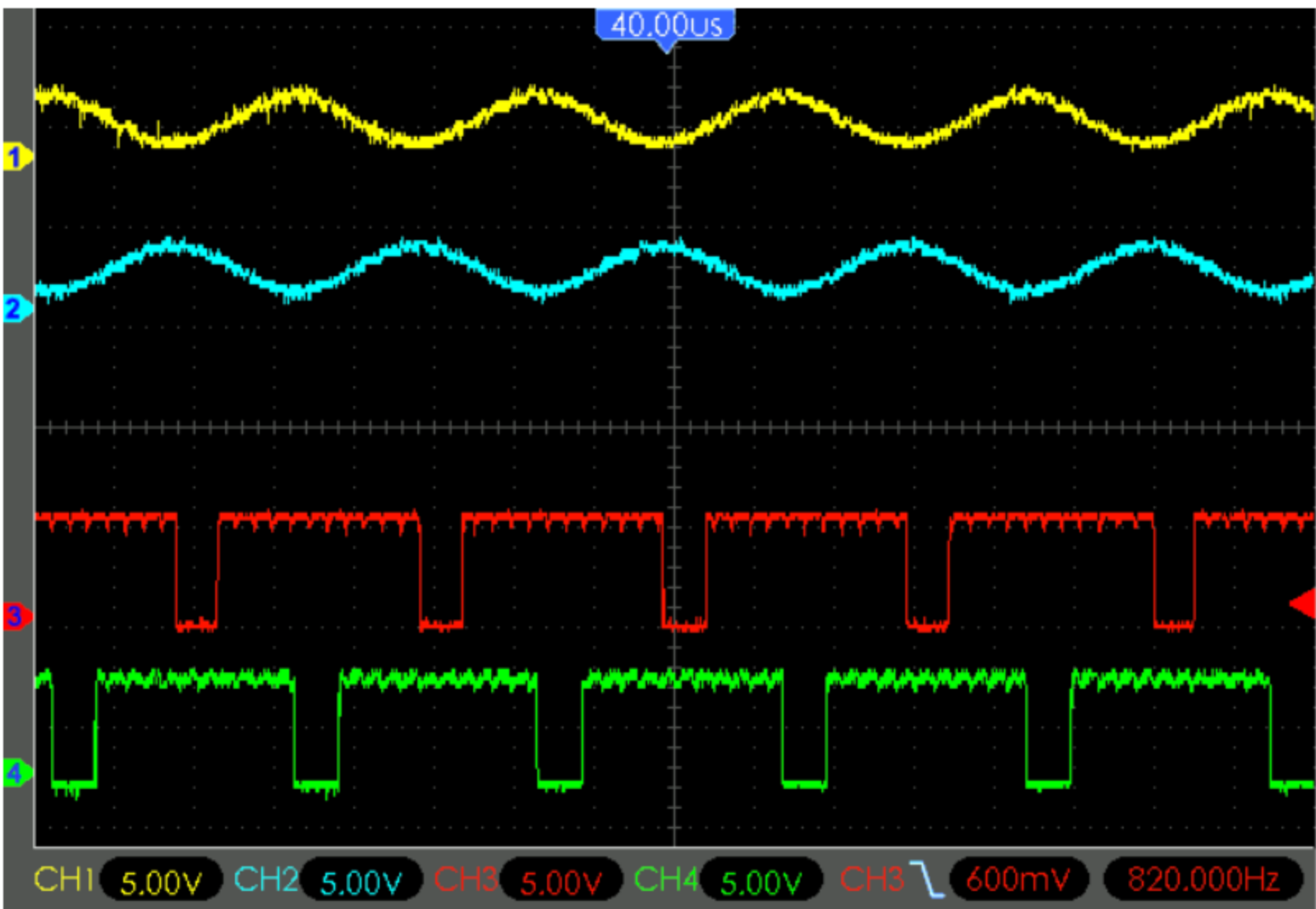
这可以说是 Speeduino 正常工作最重要的传感器。传输到 Arduino 的信号必须是 0v-5v 的方波脉冲序列（如下所示），代表以曲轴（或凸轮）速度运转的齿轮上的齿数。许多霍尔传感器和“光电”传感器都符合此数字方波规范。如果仅使用曲轴触发轮（无凸轮信号），则曲轴轮必须“缺失”一个齿才能提供位置信息以及发动机转速。目前已测试的缺齿齿轮型号有 4-1、12-1、36-1 和 60-2。

有关所有时间信号解析选项，请参阅[触发模式和解码器](#)页面。

或者（全顺序喷射的必要条件）添加带或不带缺齿曲轴轮的凸轮信号。这些设置用附加的“/x”表示，例如 60-2/1，表示 60 齿曲轴轮，缺 2 齿，每个循环产生 1 齿凸轮信号。凸轮速度缺齿曲轴轮也可以支持半顺序喷射和全顺序喷射。

也可以使用 VR（可变磁阻）传感器，但由于开发板不包含任何信号调节器来将正弦波（如下图）转换为所需的方波，因此需要一个额外的模块。v0.3.x 和 v0.4.x 系列官方开发板上有一个 8 针 DIP 插座，用于此用途，编号为 IC3。MAX9926 芯片已测试可与大多数类型的输入信号配合使用，可从[Speeduino 商店](#) 购买。此外，任何输出 0v-5v 方波的类似模块（LM1815、LM358、SSC/DSC、许多 OEM 模块等）也应该可以与 VR 传感器信号配合使用。

VR 调节器需要根据您使用的调节器，在 TunerStudio 中将触发边缘 (Trigger Edge) 设置为正确的选项。MAX 或 LM 调节器设置为上升沿 (Rising)，DSC 和大多数其他调节器设置为下降沿 (Falling)。



建议在为曲轴和凸轮轴传感器布线和选择电线时，采取预防措施以防电磁干扰 (EMI)。将电线远离交流发电机和火花塞等会产生电噪声的部件会有所帮助。使用屏蔽电缆也有帮助（将屏蔽层仅连接到 ECU 侧）。可以使用软件滤波，但没有什么比来自源头的干净信号更好。

如果曲轴/凸轮信号丢失，且 EMI/滤波选项已用尽，请确保 Speeduino 开发板的 +12V 和 GND 连接也无噪声。主电源轨上的噪声可能会造成干扰。

TPSA


TPS 传感器必须是三线电位器类型，而不是某些油门上的两线开关。如果您的 TPS 是三线传感器，那么它很可能可以工作，但您需要确认它是电位器（变量）类型传感器。

TPS 的功能是向 Speeduino 发送模拟可变电压信号，以报告油门的当前位置。它通常由 5V 的 V+ 和地（GND、信号地或信号返回）供电，并通过内部电位器进行路由，在油门开度较低时输出低电压，在油门开度较大时输出上升电压。

如果使用的TPS连接未知，建议使用欧姆表测试TPS，以确定每个引脚的连接情况，避免因随意给传感器供电而造成损坏。这可以在工作台上进行，也可以在发动机关闭并断开TPS连接的情况下进行：

- 为每个引脚分配一个字母。
- 将欧姆表连接到两个引脚，并将油门从关闭（怠速）操作到全开（WOT），并记录结果。
- 找到电阻从空闲状态到全开状态变化不大的一对引脚。这两个引脚就是你的电源引脚。
- 剩下的引脚就是**信号**引脚。

- ▶ 为了确定哪个电源引脚是**V+**，哪个是**GND**，**请测试信号**引脚和一个电源引脚之间的欧姆。
- ▶ 在空闲位置；如果欧姆值低，则电源引脚为**GND**。如果欧姆值高，则电源引脚为**V+**。

大多数可用的 TPS 传感器有 3 个引脚。如果您的 TPS 引脚数不同，请参考原始发动机接线图，了解其功能、是否可用或 Speeduino 应使用哪些引脚。对于“反向”工作且无法更改接线的 TP 传感器，[论坛](#)  上提供了一个简单的代码修改方法，以便使用此类 TPS。

MAP（歧管压力）

推荐的 MAP 传感器是飞思卡尔的 MPX4250，它集成在大多数 Speeduino 开发板上。该传感器支持高达 250kPa（1.5 bar 或约 20psi 增压）的读数，对于运行中低增压量的 NA 和增压应用来说，它是一款不错的中档传感器。

不过，其他 MAP 传感器也受支持，任何 0-5V 传感器均可使用。如果您想使用未包含在列表中的传感器（在 TunerStudio 的“工具”->“地图校准”下），请在论坛中创建一个新主题请求此操作。其他传感器也可以正常工作，但您需要在 TunerStudio 中使用不同的值进行校准。MAF 不受支持。

温度传感器（CLT 和 IAT）

任何标准的两线热敏电阻传感器均可用于这些温度功能。这些传感器的一端接地（最好来自ECU），另一端连接至信号线。这些传感器没有极性，因此这些导线的方向无关紧要。

有关详细信息，请参阅[传感器校准](#)页面

废气氧传感器（O2 和 WB02）

必须在 TunerStudio 中的工具 > 校准 AFR 表下选择 O2 传感器的类型（窄带或宽带）。

窄带

Speeduino 直接读取 NBO2 传感器信号。TunerStudio 在校准时自动为所有标准 NBO2 传感器应用标准非线性 0 伏至 1 伏值。校准后，Speeduino 将使用指定的 NBO2 根据您在 AFR 表（*Tuning > AFR Table*）中输入的内容**调整燃料**，并在*Tuning > AFR/O2*下选择传感器的类型和参数（或禁用）。请注意，窄带传感器最初设计为针对化学计量 AFR（Lambda 1.0）以实现高效的催化排放控制，并且通常不够准确或不适合调整高效的稀薄经济性或浓厚动力燃料。虽然不推荐；但可以使用复杂的调整方法，使用 NBO2 传感器对稀薄和浓厚 AFR 进行有限和近似的调整。

宽带

宽带氧传感器 (WB02) 可以检测并报告比窄带传感器更宽的 λ (λ) 或空燃比 (AFR) 范围，并且精度更高，范围约为 10:1 到 20:1（约 0.7 到 1.3 λ ），具体取决于具体的传感器版本和控制器。Speeduino 无法直接使用 WB02 传感器，需要外部控制器来处理信号并应用传感器加热控制。请从显示的列表中输入控制器品牌和型号。如果控制器信号是通用线性或自定义的，请选择并输入所需信息，或者在菜单列表中选择安装自定义 INC 文件。

在“工具”>“校准 AFR 传感器”中设置后，Speeduino 可以使用指定的 WB02 将 lambda/AFR 报告给 TunerStudio 以供仪表显示。在“调节”>“AFR/O2”下选择传感器的类型和参数后，它可以根据您在 AFR 表（“调节”>“AFR 表”）中输入的内容，动态调整校正燃油，并在 TunerStudio 或 MegaLogViewer 中实时或从日志中进行自动调节。设置中还包含禁用选项。虽然 Speeduino 可以使用 WB02 信息来校正燃油，但强烈建议不要使用它来补偿调节不当。

特定应用输入

Speeduino 用户发现电路和技术对于调整或实现某些输入或功能很有用。

柔性燃料传感器

有关灵活燃料设置的硬件和配置的详细信息，请参阅[灵活燃料部分](#)。

12V输入信号

一些位置传感器输出 12V 信号。为了解决这个问题并避免损坏 Arduino，可以构建一个如图所示的电路。电阻 R1 并非必需，但它可以确保任何非高电平输出都被拉低。配合此电路使用 Speeduino 上的上拉跳线。这样可以有效地将 0v/12v 信号转换为 0v/5v 信号。

非常感谢 PSIG 提供的信息和图表。


GM 7/8针分配器模块

请参阅[GM_Module](#)了解更多详情

输出

有关每个输出应如何连接的详细信息，请参阅菜单中的接线图链接。

喷油器

Speeduino 喷油器驱动器使用开/关（而非 PWM）控制，设计用于与“高阻抗”喷油器配合使用。这类喷油器也称为“饱和”或“高阻抗”喷油器，它们使用满电池电压来控制喷油器的开启循环，其阻抗通常大于 8 欧姆。如果您使用的是阻抗较低的“低阻抗”（峰值保持或 PWM 控制）喷油器，则需要在它们上安装串联电阻，以避免过大电流损坏电路板。电阻的欧姆数和额定功率可以根据欧姆定律计算，或者使用在线计算器页面，例如[Speeduino 喷油器电阻计算器](#) 。

Speeduino 每个输出通道最多可驱动 2 个 High-Z 喷射器。

0.4V 电路板每个喷油器通道有 2 个引脚，可直接将每个喷油器连接到电路板，无需使用线束接头。如果应用中包含的喷油器少于 4 个，则使用连接到相应通道的任意引脚即可。

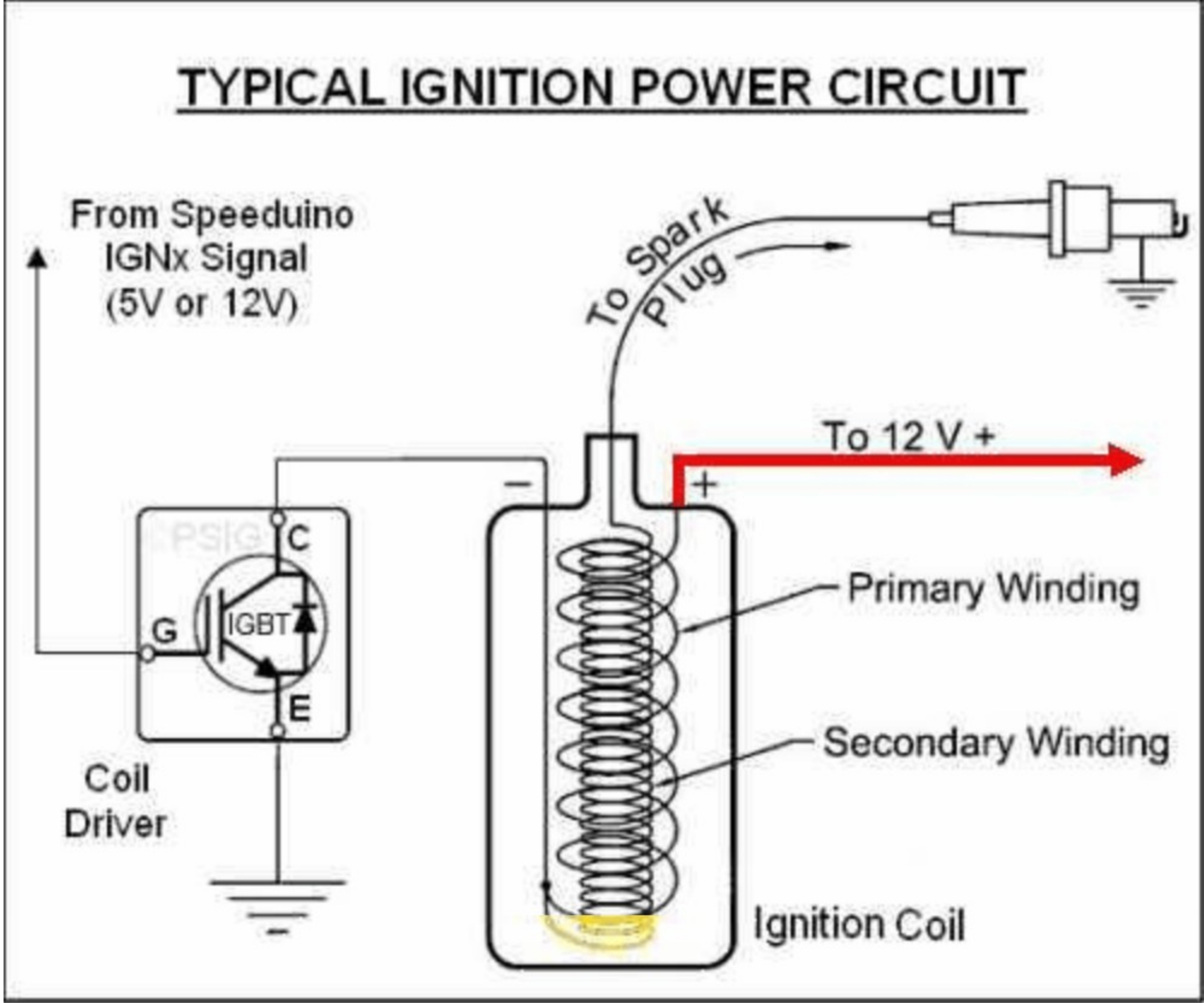
线圈

Speeduino 的当前版本使用低功率输出信号，旨在与外部小信号点火线圈驱动器配合使用，无论是独立式（模块或 ICM、点火器、IGBT 等），还是内置于线圈组件中（“智能”线圈）。这种方法使 Speeduino 能够灵活地控制大多数类型的点火系统。如果将 Speeduino 输出直接连接到传统的高电流无源（“哑”或 2 针）点火线圈（不使用点火线圈驱动器），则会损坏您的 Arduino。

Speeduino 如何控制点火电路功率以往，线圈驱动器是一组机械接触点（“点”），如今已被电子驱动器取代。新增的线圈驱动器可以位于 Speeduino 内部或线圈组件内部的任何位置；靠近或位于线圈内部可以降低电气噪声。

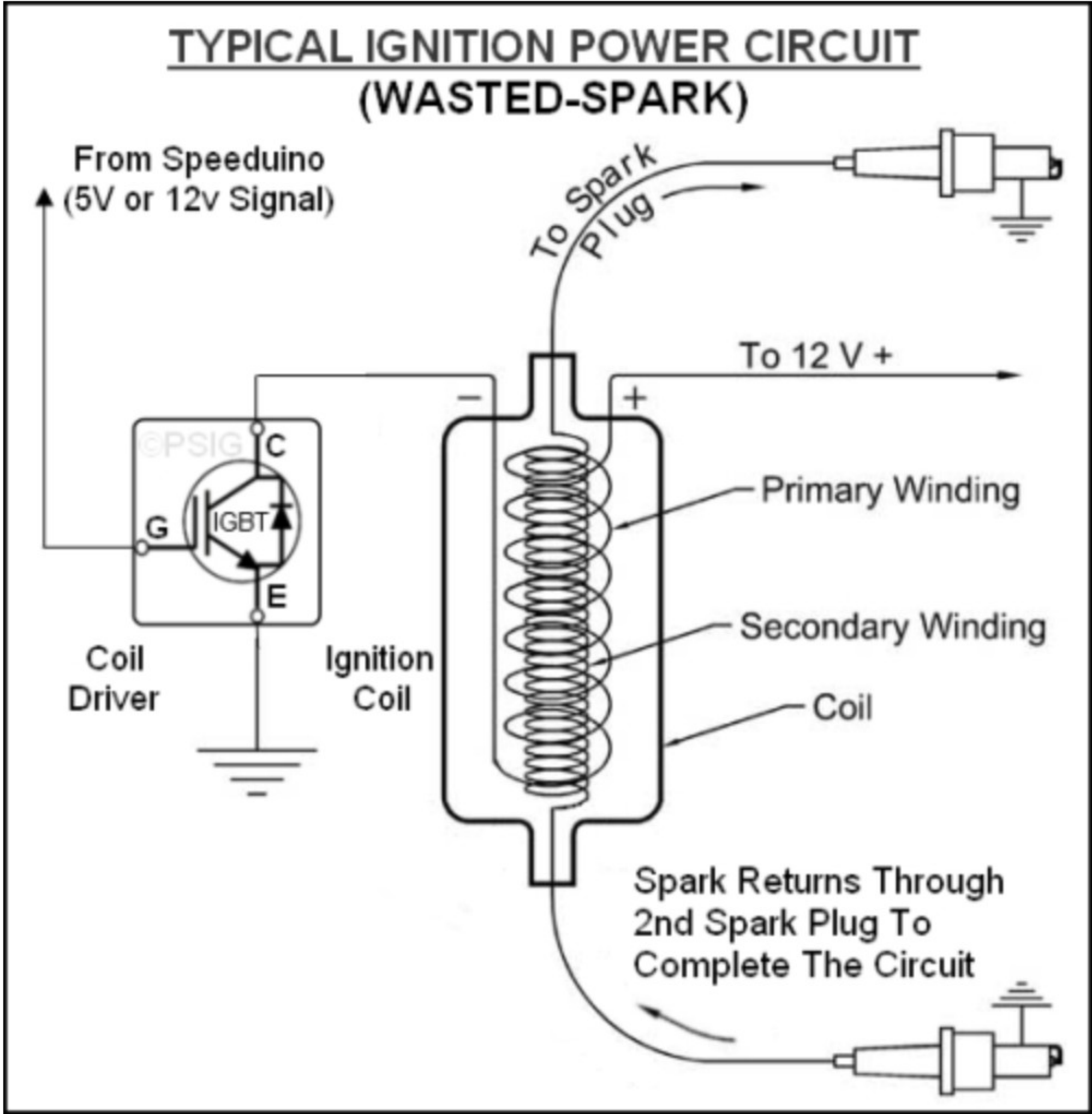
在下方的示例动画中，Speeduino 点火信号通常为低电平（接近地或 0V），同时 Speeduino 计算下一次线圈点火的时间。在适当的时间，Speeduino 将点火输出切换至高电平（接近 5V 或 12V，可选），以开启线圈驱动器（例如 IGBT），使电流流经线圈并接地。这被称为“闭合”期。在闭合期间，点火线圈磁芯和线圈绕组周围会产生一个不断增强的能量场。

在停留时间结束时，也就是火花产生的适当时间；Speeduino 会关闭线圈驱动器，停止电流，从而使能量场崩溃，产生高电压并产生火花：



在 TunerStudio 中，本例的设置是“低速”点火。闭合角设置会根据线圈类型、电压等级等因素而变化很大。闭合角过小会导致火花微弱或无火花，而闭合角过大则会迅速升温，可能损坏任何点火组件，但通常会损坏点火线圈或线圈驱动器。

下面是浪费火花的版本，显示了其操作方式相同，但高压火花通过第二个火花塞返回以完成电路：




关于“智能”线圈类型的详细列表可参见：<http://www.megamanual.com/seq/coils.htm>。Speeduino 提供多种点火模块，可用于控制标准线圈；对于智能线圈，通常可以使用 4 针或 5 针类型，因为它们始终为逻辑电平，尽管一些 3 针线圈也属于此类。GM LS1/2 线圈是常用的强大智能线圈的一个例子，通常可以轻松且廉价地获得。

（注：过去，一些具有限流或闭合控制功能的点火控制模块（例如，20 世纪 70 年代的通用 HEI、博世的“024”型和福特 DS1）被称为“智能”模块。虽然现在仍然如此，但至少集成一个驱动器的独立点火线圈，或具有更强大控制功能的新技术（例如，控制火花持续时间或多火花）都被视为“智能”线圈。您必须了解您打算使用的特定驱动器、控制模块或线圈的控制要求，才能使用 Speeduino 正确操作它们。）

辅助输出

低电流

大多数 Speeduino 版本都具有低电流 (LC) 信号输出，这些信号直接从 MEGA 处理器输出到（通常）开发板的原型 (proto) 区域。这些输出通常不适合以这种形式直接控制电源设备，需要在原型区域构建合适的输出电路才能使用。或者，燃油泵或风扇等输出功能可以重新分配给其他引脚和组件，例如现有的高电流 (HC) 输出 MOSFET——更多信息请参阅下文的“中电流”部分。

部分 Speeduino 版本包含 8 通道 ULN2803A 达林顿晶体管阵列 IC，每通道最高可切换 0.5 安培电流。这些辅助输出足以直接切换小型设备，或切换功率 MOSFET 或汽车继电器等高功率器件。这些输出的配置和设置在“[配置](#)”部分有详细说明。有关在 v0.4 系列开发板上使用 ULN2803A 的更多信息，请参阅[此处](#) 。其他开发板设计中可以使用类似的输出选项和引脚分配。

中等电流


v0.3x 及更高版本的电路板包含中等功率 MOSFET 辅助输出，可直接切换高达 3 安培的电流。这些输出在原理图和一些参考资料中标记为“HC”。这些输出通常用于操作怠速阀、增压控制阀、可变气门正时 (VVT) 电磁阀等，或控制继电器以处理更大的负载，例如电动散热器风扇。这些输出的配置和设置将在“[配置](#)”部分中描述。

辅助输入输出

[Speeduino 通过Serial3](#)支持最多 16 个 16 位外部模拟输入

CAN总线


如果使用 STM32 或 Teensy MCU，Speeduino 可直接使用 CAN 总线；如果使用 mega2560，则可通过单独的协处理器使用。在官方开发板上，[Dropbear](#) 是目前唯一一款支持 CAN 总线的开发板，它使用 Teensy 3.5 MCU。但也有第三方设计支持 CAN 总线。更多关于 Speeduino CAN 总线支持的信息，请点击[此处](#)。

由于 Arduino mega2560 没有 CAN 总线接口，因此设计了一个单独的“协处理器”接口。更多关于此设备的信息可以[在这里](#)  找到。它使用 Serial3 端口提供的功能并通过该端口进行连接。

第三方插件

在本节中，您将找到有关与 Speeduino ECU 配合使用的第三方设计硬件的信息

Speeduino 的 GPIO

GPIO 有几种变体，固件可以[在这里下载](#) 

更多信息[GPIO_for_Speeduino](#)