


[店](#)[学习](#)[博客](#)[服务](#)

配电板-USB-C (Qwiic) 连接指南

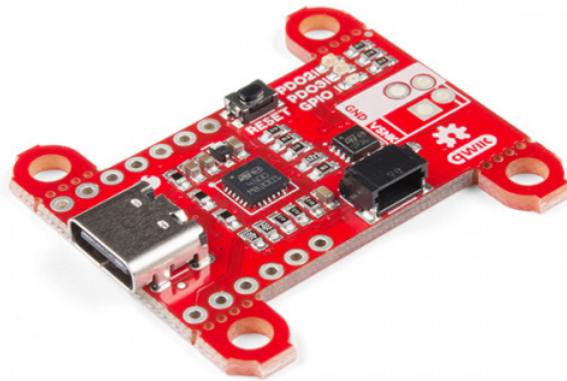
贡献者:  亚历克斯巨人, ELL C

♡ 喜爱

1个

介绍

USB Type-C给USB标准带来了两个重大变化。可逆连接器消除了每次尝试以正确方式插入连接器的问题。第二个主要变化是允许灵活性,使USB电源具有可调节的USB电压,电压范围为5V至20V,最高可提供100W的功率。该SparkFun电力输送局充分利用了电力输送的标准与使用独立控制器的意法半导体的STUSB4500。控制器完成了电源协商的所有繁重工作,并提供了一种通过I²C进行配置的简便方法。



英国威廉希尔SparkFun电力输送板-USB-C (Qwiic)

DEV-15801

\$ 24.95

★★★★☆ 2

Product Showcase: SparkFun Power Delivery B...



所需材料

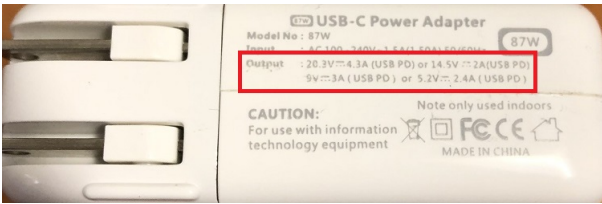
要遵循本指南，您将需要以下材料。您可能并不需要一切，这取决于您拥有什么。将其添加到购物车，通读指南，并根据需要调整购物车。

配电板USB-C的愿望清单 SparkFun的愿望清单	
	英国威廉希尔SparkFun电力输送板-USB-C (Qwiic) DEV-15801
	Qwiic电缆-100mm PRT-14427
	英国威廉希尔SparkFun坚固的microB电缆-1m CAB-14742
	英国威廉希尔SparkFun RedBoard Qwiic DEV-15123

其他工具

您还需要支持电源传输的USB C型电源适配器和电源C型电缆，电缆中的电源线较粗。这样可以最大程度地减小电缆两端的压降，并在大电流负载下发热更少。供电板的电压和可用电流输出受所用供电适配器的限制。本指南中使用的电缆和电源适配器是Apple制造的87W USB-C电源适配器，并支持5.2V，9V，14.5V和20.3V。

⚡ **注意：**在测试过程中，我们注意到某些电源适配器同时具有USB Type-A快速充电插头和USB Type-C供电插头，在改变电压方面存在明显的延迟。我们建议您获取仅支持USB Type-C连接器的电源适配器。



电源适配器输出电压

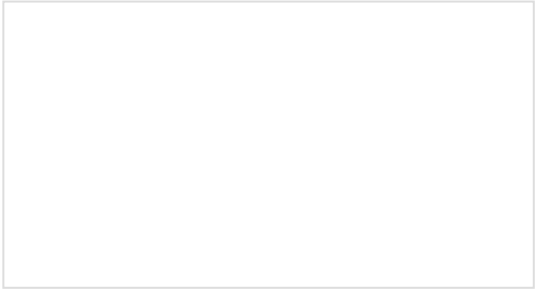
建议阅读

在继续阅读本指南之前，您可能需要熟悉其中一些不熟悉的主题。如果您不熟悉Qwiic系统，建议阅读此处以获取概述。

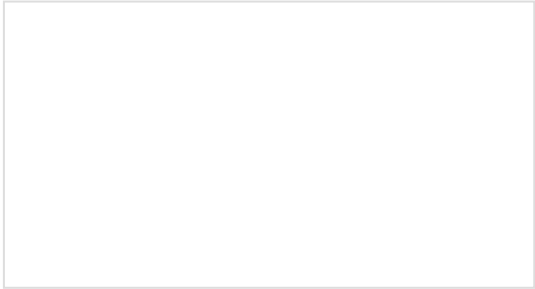


Qwiic Connect系统

如果您不熟悉以下概念，我们建议您先阅读这些教程，然后再继续。在上传代码之前，请确保安装适当的驱动程序。在这种情况下，我们需要确保为RedBoard Qwiic安装了CH340驱动程序。



I2C
I2C简介，I2C是当今使用的主要嵌入式通信协议之一。



如何使用万用表
了解使用万用表测量连续性，电压，电阻和电流的基础知识。



RedBoard Qwiic联播指南
本教程介绍了RedBoard Qwiic的基本功能。本教程还介绍了如何开始闪烁LED和使用Qwiic系统。



如何安装CH340驱动程序
如何在Windows，Mac OS X和Linux上安装CH340驱动程序（如果需要）。

硬件总览

STUSB4500

在电力输送方面，您有两种类型的电力角色。第一个是提供程序，也称为源，它能够通过USB电源总线提供电源。另一个角色是消费者，也称为接收器，它从源头汲取功率。

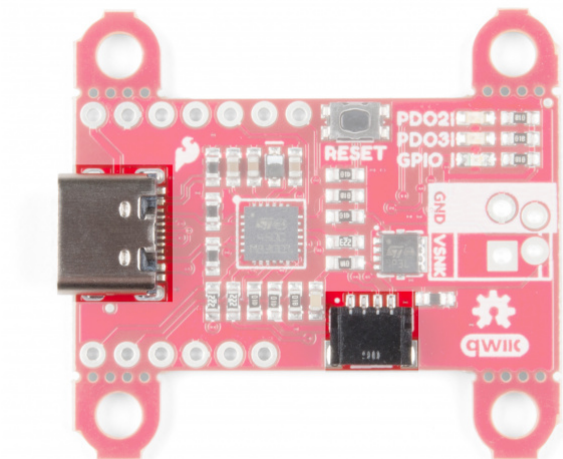
STUSB4500是用于接收器应用的USB Type-C和电源输送控制器IC。尽管您将需要一个微控制器来配置该板，但该控制器无需外部微控制器就可以与电源（即，壁装电源适配器或电源适配器）协商电源合同。通过该控制器，我们可以利用意法半导体的专有算法和使用集成非易失性存储器（NVM）的可配置电源数据对象（PDO）。其洗衣功能列表包括：

- 两个USB C型端口之间的“连接检测”
- 建立有效的源到汇连接
- 与支持PD的电源协商USB供电（PD）合同
- 监视VBUS电源路径并管理VBUS电压转换
- 管理高压保护
- 双高功率充电电路支持

了解如何在USB-C电源协商部分中讨论电源协商，但是现在，让我们来看一下电路板本身。

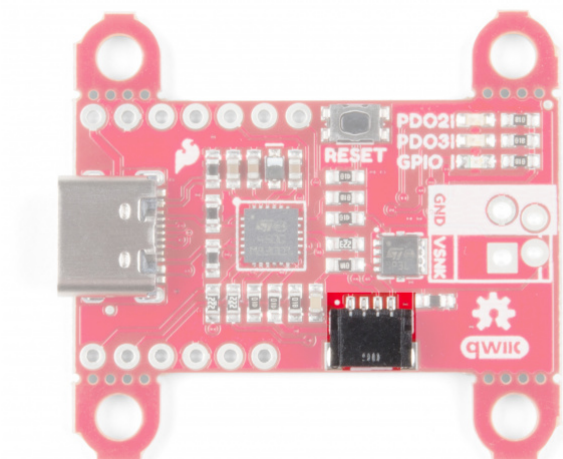
功率

可以通过USB Type-C连接器或通过Qwiic连接器提供的3.3V两种方式Power Delivery板供电，如果您只需要配置新的Power Delivery配置文件。

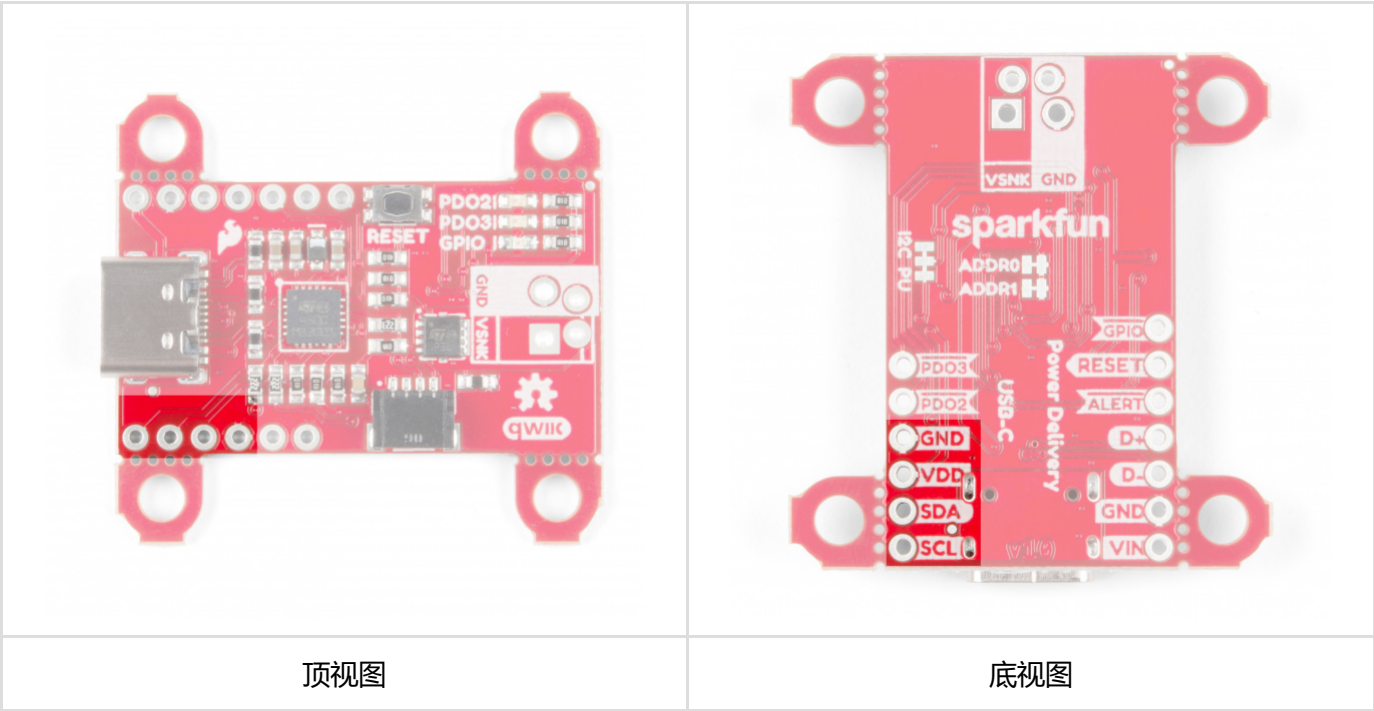


I²C引脚

要配置该板，您将需要I²C总线。Qwiic系统使将电源传输板连接到微控制器变得很容易，以设置NVM参数以通过Qwiic连接器为您的项目供电。根据您的应用，您还可以通过SDA和SCL的镀通孔连接到I²C总线。



Qwiic连接器



I2C PTH引脚

别针

该板分出了以下引脚：

引脚名称	描述
VIN	这是输入电源引脚，连接到C型连接器上的VBUS。
VSNK	成功进行电源协商后启用电源输出
VDD	可选逻辑电源输入， 3.0V-5.5V
地线	地面
SCL	I ² C时钟输入
SDA	I ² C数据输入/输出
D-	USB 2.0差分对， 负极
D +	USB 2.0差分对， 正极
警报	I ² C中断， 低电平有效开漏
重启	复位输入， 高电平有效
通用输入输出	通用输出， 低电平有效开漏
PDO2	PDO2的电源合同标志， 低电平有效开漏

PDO3	PDO3的电源合同标志，低电平有效开漏
------	---------------------

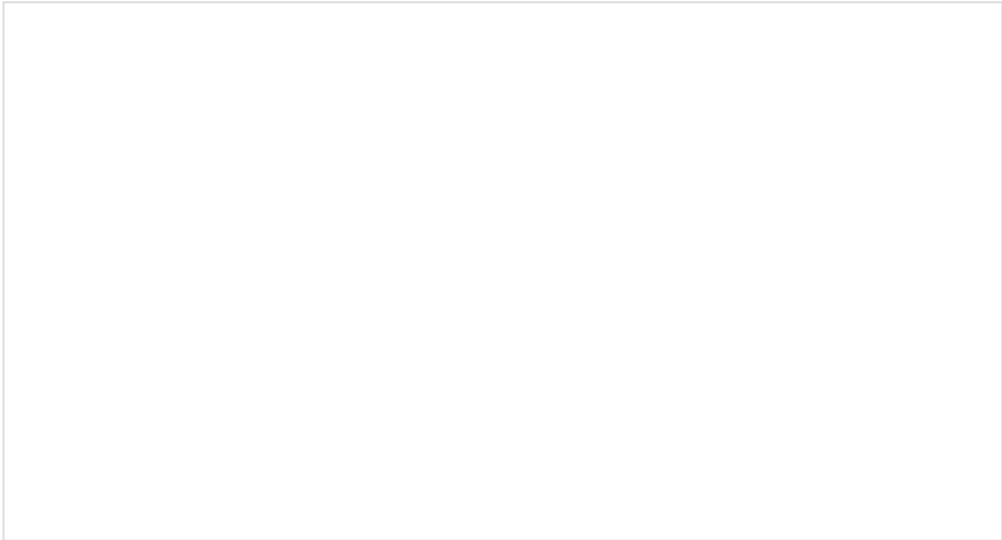


IO引脚

跳线

该板上有几个跳线垫，用于配置I²C总线。

i 如果您尚未使用跳线垫，请确保查看有关如何使用跳线垫和PCB走线的教程以获取更多信息。



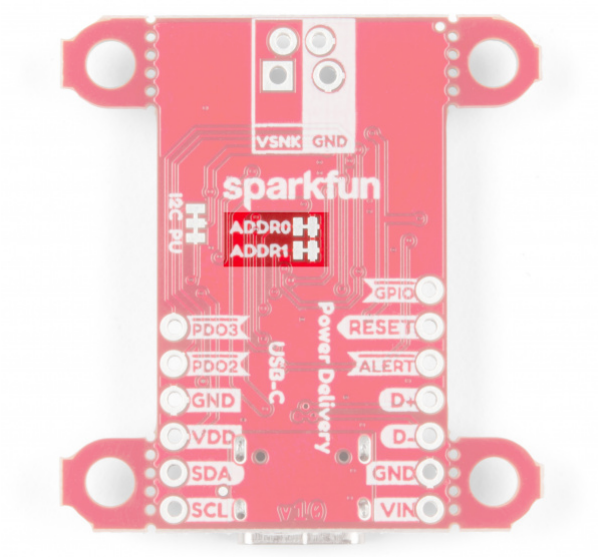
如何使用跳线垫和PCB走线

2018年4月2日

处理PCB跳线焊盘和走线是必不可少的技能。了解如何切割PCB走线，在焊盘之间添加焊料跳线以重新布线，以及在走线损坏时使用绿线方法修复走线。

地址选择

单板的默认地址为**0x28**。如果需要调整电路板的地址，则可以剪切地址标签**ADDR0**和**ADDR1**中的一个或两个。

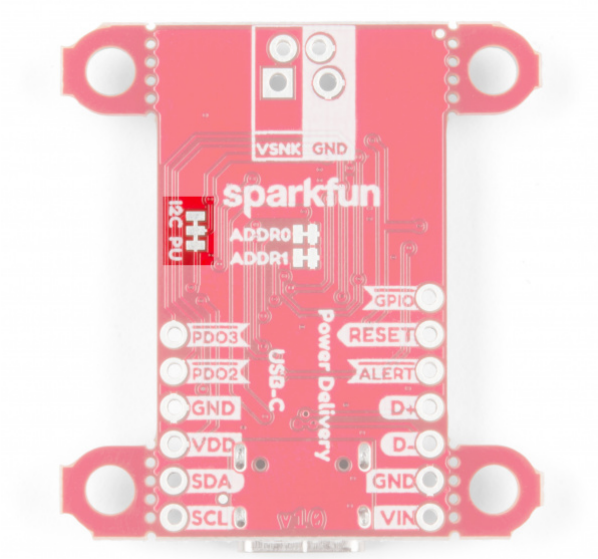


下表列出了四个跳线配置以及相应的十六进制设备地址。

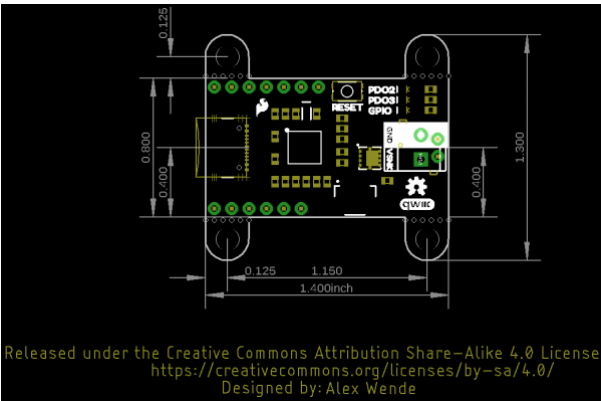
地址0	地址1	设备地址
关闭	关闭	0x28 (默认)
打开	关闭	0x29
关闭	打开	0x2A
打开	打开	0x2B

上拉电阻

该评估板还包括跳线，用于禁用I²C总线上的上拉电阻。如果您在同一总线上使用几个I²C器件，而这些I²C器件的板上已经装有上拉电阻，则可能需要切断跳线以断开这些上拉电阻。Power Delivery板利用非易失性内存，即使断电，保存到设备的设置仍会保持配置状态，因此，该板需要通过I²C连接的唯一时间是修改NVM参数。



外型尺寸



USB-C电源协商

电源数据对象（PDO）

STUSB4500具有一组用户定义的参数，可以通过I²C接口使用NVM重新编程来自定义这些参数。这允许更改供电接口的预配置，并根据特定的应用要求定义新的配置。

上电或复位后，NVM参数被复制到I²C寄存器中，并由控制器在系统运行期间使用。控制器最多可以存储三个PDO，其中PDO3具有最高优先级，而PDO1具有最低优先级。将NVM参数复制到I²C寄存器后，STUSB4500将询问源其查找PDO匹配项的能力。

每个PDO都有四个参数，如下表所示：

参数	取值范围	笔记
电压	5至20 (V)	PDO1固定为5V，无法更改。
过电压容差	5-20 (%)	设置上限电压公差。例如，假设此参数设置为5%，电压设置为9V。如果电源电压超过9.45V，则输出（VSNK）将关闭。
欠压容差	5-20 (%)	仅为PDO2和PDO3设置较低的电压容限。例如，假设此参数设置为5%，电压设置为9V。如果电源电压降至8.55V以下，则输出（VSNK）将关闭。 PDO1的固定欠压限制为3.3V，无法更改。
当前	0-5 (A)	16个可能的值，请参阅 setCurrent Arduino库中的函数以获取确切值。

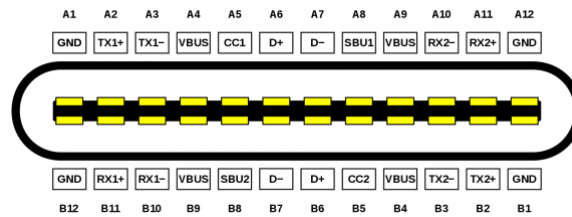
每个电源在外壳上列出的电压可能略有不同。一种充电器可能会标出12V，而另一种充电器可能标出12.3V，也许是为了补偿电缆两端的电压降，而另一种充电器可能标出11.5V，以指示满载情况下电缆末端的电压。不过，大多数来源在与消费者协商时都会宣传5V，9V，12V，15V或20V的某种组合。

电流应该是电源输出板期望从电源汲取的最大电流量。如果当前参数小于或等于广告源所宣传的最大电流，则控制器将向电源发送该功率曲线的请求。如果电压或电流参数与电源的功能之一不匹配，则控制器将尝试下一个PDO查找与电源的匹配。

如果源接受合同，则源将从默认的VBUS电压5V切换到请求的电压，并且VBUS_EN_SINK引脚将被拉低，并使电流流过MOSFET，该MOSFET控制功率到VSNK电源输出引脚。

源和宿之间的电力输送合同

USB中的“U”代表通用，当连接两个设备时，在它们之间会发送大量消息，以确定每个设备连接的对象以及每个设备支持的内容。为了简单起见，我们仅从较高的角度看待电源供应合同，以了解发生了什么情况。



图片由 Chindi.ap提供, CC BY-SA 4.0

在USB Type-C连接器的24个引脚（如上所示）中，STUSB4500仅直接连接至其中的10个引脚：4个引脚用于VBUS，另外4个引脚接地，以及2通道配置或CC引脚。根据电缆的方向，这些CC引脚之一用于在接收器和源之间发送和接收消息。当宿设备连接到电源时，VBUS上的电压开始于5V，就像旧的USB标准一样。

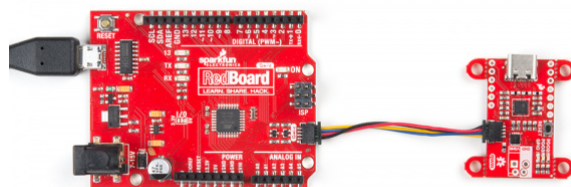
连接电源和接收器后不久，电源将宣传其作为电源的功能，例如5V @ 3A，9V @ 2A以及它能够提供的任何其他电源传送选项。收到该消息后，控制器将查看可用于查找匹配项的PDO选项。匹配后，控制器将询问电压和电流选项之一。电源接受后，电压便切换到要求的电压。

为了找到匹配项，STUSB4500控制器首先查看SNK_PDO_NUMB参数，该参数是1到3之间的整数值，它对应于最高优先级的PDO编号。如果SNK_PDO_NUMB的值为3，它将首先检查源是否能够提供PDO3，否则将检查PDO2，然后检查PDO1。如果SNK_PDO_NUMB的值为2，它将首先检查PDO2，然后检查PDO1，然后忽略PDO3。最后，如果SNK_PDO_NUMB的值为1，它将仅检查PDO1。控制器寻找匹配的两个值是电压和电流。如前所述，大多数电源适配器将具有5V，9V，12V，15V和20V的组合。包装上可能说的是5.2V或20.3V，但是在功能消息中，它们实际上将指定5V或20V。

合同的另一部分是当前要求的合同。如果电压被接受，但只能在该电压下提供1.5A的电流，但要求2.0A的电流，则即使电源能够提供该电压，该合同也将被电源拒绝。或者，将接受能够提供1.5A电源的0.5A电流请求。请求的电流应约为该项目的预期值，但电源将允许接收器汲取比请求更多的电流，直至该功率传输选项的限制。

硬件连接

这是一个基于I²C的板，这使我们能够在分线板上包括Qwiic连接器。连接设备很容易，只需将Qwiic电缆的一端插入电源输出板，另一端插入开发板即可。在这种情况下，它是RedBoard Qwiic。



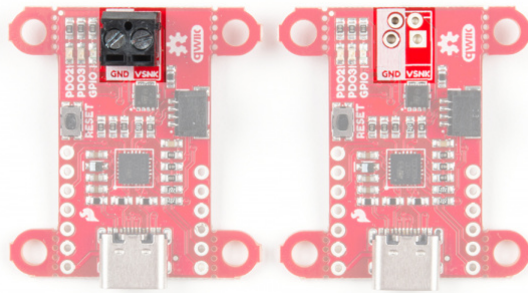
注意：如果您以前从未将CH340设备连接到计算机，则可能需要安装USB到串行转换器的驱动程序。请参阅“如何安装CH340驱动程序”一节以获取安装帮助。

如何安装CH340驱动程序

2019年8月6日

如何在Windows, Mac OS X和Linux上安装CH340驱动程序（如果需要）。

为项目提供电源可以通过两种方式来完成。该评估板可提供面包板友好的0.1英寸间距以及3.5mm的螺钉端子间距。如果您不熟悉焊接，请参阅我们的“如何焊接指南”。



配置好板卡后，可以卸下Qwiic连接器，即使断开电源，板卡也会记住设置。有了这些连接，就可以配置电路板了！

Arduino库

注意：本示例假定您在桌面上使用最新版本的Arduino IDE。如果这是您第一次使用Arduino，请查看我们的有关安装Arduino IDE的教程。如果您以前尚未安装Arduino库，请查阅我们的安装指南。

英国威廉希尔SparkFun编写了一个库，用于控制STUSB4500的电源传输板。您可以通过Arduino Library Manager获取该库。通过搜索**STUSB4500**，您应该会看到由**SparkFun Electronics**编写的书，并且应该能够安装最新版本。如果您希望手动下载库，则可以从GitHub Repository中获取它们。

下载SPARKFUN电源传送板-USB-C ARDUINO库 (ZIP)

职能

该库具有大量功能，可用于读取Power Delivery板的每个NVM参数。以下是可用功能的列表，以及每个功能的用途以及使用方法的说明。

- **uint8_t begin (uint8_t deviceAddress, TwoWire & wirePort)** -在草图的开头调用以初始化设备。此函数采用两个可选参数：deviceAddress和wirePort。true 初始化设备后，false 如果无法与设备通信，此功能将返回。
 - **wirePort**-默认情况下，该库使用标准的Wire总线，但是如果您的电路板具有多个I2C总线，则可以在此处定义Wire1。
 - **deviceAddress**-如果板子背面的两个地址跳线均已关闭，则无需提供此参数。否则，您可以使用下表获取电源传输板的I²C地址。

地址0	地址1	设备地址
关闭	关闭	0x28 (默认)
打开	关闭	0x29
关闭	打开	0x2A
打开	打开	0x2B

- **void read (void)** -此函数读取NVM参数并将其加载到内存中。该函数在函数中被调用，begin() 不需要在程序中被调用。
- **void write (uint8_t defaultVals)** -此函数将NVM参数写入STUSB4500。更改完所有参数后，调用此函数会将参数保存到设备中。此函数采用的可选参数 DEFAULT。将DEFAULT传递给写入函数时，会将默认的NVM参数写入设备。
- **float getVoltage (uint8_t pdo_num)** -返回为PDO编号 (1-3) 请求的电压。
- **float getCurrent (uint8_t pdo_num)** -返回当前请求的PDO编号 (1-3) 。
- **uint8_t getLowerVoltageLimit (uint8_t pdo_num)** -返回PDO编号 (1-3) 的欠压锁定参数。PDO1 具有无法更改的固定值，并将始终返回0。欠压限制固定为3.3V。PDO2和PDO3将返回5%到20%之间的值。
- **uint8_t getUpperVoltageLimit (uint8_t pdo_num)** -返回PDO编号 (1-3) 的过压锁定参数。返回的值将介于5%和20%之间。
- **float getFlexCurrent (void)** -返回flex current参数的值，该参数是所有PDO通用的全局变量。
setFlexCurrent 有关如何使用弯曲电流的更多信息，请参考功能。
- **uint8_t getPdoNumber (void)** -返回保存在内存中的优先级最高的PDO编号 (1-3) 的值。
- **uint8_t getExternalPower (void)** -返回SNK_UNCONS_POWER参数的值 (0或1) 。
SNK_UNCONS_POWER是接收器发送的功能消息中的不受约束的功率位设置。0表示没有外部电源 (默认)，1表示可以使用外部电源，足以在给外部设备充电时为系统充分供电。
- **uint8_t getUsbCommCapable (void)** -返回USB_COMM_CAPABLE参数的值 (0或1) 。
USB_COMM_CAPABLE是指接收器系统的USB2.0或3.x数据通信功能。0表示接收器设备不支持数据通信 (默认)。1表示接收器确实支持数据通信。
- **uint8_t getConfigOkGpio (void)** -返回POWER_OK_CFG参数值 (0-3) ，该参数值控制VBUS_EN_SNK，POWER_OK2和POWER_OK3引脚的行为。setConfigOKGPIO 有关更多信息，请参考该功能。
 - 0-配置1
 - 1-不适用

- 2-配置2 (默认)
- 3-配置3
- **uint8_t getGpioCtrl (void)** -返回GPIO引脚 (0-3) 的行为设置。setGpioCtrl 有关更多信息, 请参考该功能。
 - 0-软件控制的GPIO
 - 1-错误恢复
 - 2-调试
 - 3-接收功率
- **uint8_t getPowerAbove5vOnly (void)** -返回POWER_ONLY_ABOVE_5V参数 (0或1)。如果返回0, 则在连接电源时无论电压如何都将启用输出。如果返回1, 则仅在连接电源并将VBUS电压协商到PDO2或PDO3时, 才启用输出。
- **setPdoNumber (uint8_t pdo_num)** -使用参数pdo_num (1-3)。Power Delivery板可以存储三个Power Data Objects (PDO)。PDO3具有最高优先级, 其次是PDO2, 最后是PDO1。此函数声明应首先协商三个PDO中的哪个。例如, 如果setPdoNumber设置为3, 则将首先协商PDO3, 如果未签订合同, 则协商PDO2, 如果PDO2也失败, 则协商PDO1。如果setPdoNumber设置为2, 则将忽略PDO3并协商PDO2, 然后再协商PDO1。最后, 如果setPdoNumber设置为1, 则仅协商PDO1。
- **void setVoltage (uint8_t pdo_num, float voltage)** -采用两个参数: pdo_num (1-3) 和电压 (10位分辨率)。PDO1固定为5V, 无法更改。PDO2和PDO3可以是最高20V的任何电压。
 - 从 setPdoNumber() 参数中定义的数字开始, PDO3的优先级最高, 而PDO1的优先级最低。该电源控制存在于VBUS上的电压, 或者控制在电源传输板VIN上的电压。可从电源获得的常见电压为: 5V, 9V, 12V, 15V和20V。
- **void setCurrent (uint8_t pdo_num, float current)** -具有两个参数: pdo_num (1-3) 和current (16个值)。电流值为: 0 (弹性电流), 0.5A, 0.75A, 1.0A, 1.25A, 1.50A, 1.75A, 2.0A, 2.25A, 2.50A, 2.75A, 3.0A, 3.5A, 4.0A, 4.5A和5.0A。
 - 当前值应该是期望光源输出的数量。如果当前价格高于货源提供的价格, 则该合同将被拒绝, 并协商下一份PDO合同。如果电源能够供电, 则Power Delivery板将消耗的电流比协商的要多, 但是建议提供预期项目将消耗的最大电流。
- **setFlexCurrent (float value)** -取一个float值来设置所有PDO的公共电流。仅当该PDO的setCurrent值设置为0时, 才在电源协商中使用此值。柔性电流的分辨率为10mA。就像 setCurrent 函数一样, 当前值应该是源应交付的数量。如果当前价格高于货源提供的价格, 则该合同将被拒绝, 并协商下一份PDO合同。
- **setLowerVoltageLimit (uint8_t pdo_num, uint8_t value)** -采用两个参数: pdo_num (2-3) 和一个整数值 (5-20%)。由于PDO1的固定值为3.3V, 因此只能更改PDO2和PDO3。与 setUpperVoltageLimit 功能结合使用, 可用于定义电压输出的可接受窗口。
 - 例如, 如果电源输出板要求15V且欠压容差为5%, 则当电压下降到14.25V以下 (即15V的5%) 以下时, 控制器将禁用电源输出板的输出。
- **setUpperVoltageLimit (uint8_t pdo_num, uint8_t value)** -采用两个参数: pdo_num (1-3) 和一个整数值 (5-20%)。与 setLowerVoltageLimit 功能结合使用, 可用于定义电压输出的可接受窗口。
 - 例如, 如果电源输出板要求15V且具有5%的过压容限, 则当电压超过15.75V (即15V的5%) 时, 控制器将禁用电源输出板的输出。
- **setExternalPower (uint8_t value)** -取一个整数值 (0或1)。设置为0时, 没有外部电源。设置为1表示有外部电源可用, 足以在为外部设备充电时为系统充分供电。
- **setUsbCommCapable (uint8_t value)** -取一个整数值 (0或1)。设置为0, 则接收器不支持数据通信。设置为1表示接收器确实支持数据通信。
- **setConfigOkGpio (uint8_t value)** -取一个整数值 (0-3) :
 - 0-配置1

- 1-不适用
- 2-配置2 (默认)
- 3-配置3

	销	州	描述
配置 1	VBUS_EN_SNK	高 Z	没有来源
		0	附带来源
	POWER_OK2	高 Z	没有功能
	POWER_OK3	高 Z	没有功能
配置 2	VBUS_EN_SNK	高 Z	没有来源
		0	附带来源
	POWER_OK2	高 Z	没有明确的PD合同
		0	与PDO2的PD明确合同
	POWER_OK3	高 Z	没有明确的PD合同
		0	与PDO3的PD明确合同
配置 3	VBUS_EN_SNK	高 Z	没有来源
		0	附带来源
	POWER_OK2	高 Z	连接电源时, 没有电源连接或电源提供5V的默认USB Type-C电流。
		0	连接电源时, 电源以5V提供3.0A USB Type-C电流。
	POWER_OK3	高 Z	连接电源时, 没有电源连接或电源提供5V的默认USB Type-C电流。
		0	连接电源时, 电源以5V提供1.5A USB Type-C电流。

*

注意：对于配置3：“源供给1.5A / 3.0A USB C型电流在5V时源被附接”是基于什么源通告当电缆连接，并且也未指示输出电压实际上是5V。

- **setGpioCtrl (uint8_t value)** -取一个整数值 (0-3) 来控制GPIO引脚 (低电平有效)。

值	GPIO功能	功能	值	描述
0	软件控制的GPIO	输出状态由存储在I ² C寄存器0x2D的bit-0中的值控制	高 Z	当bit-0的值为0时 (启动时的默认值)
			0	当bit-0的值为1时
1 个	错误恢复	硬件故障检测，例如温度过高，CC引脚上过电压或硬复位后	高 Z	未检测到硬件故障
			0	检测到硬件故障
2	除错	调试附件检测 (有关更多信息，请参见数据表第3.8节)	高 Z	未检测到调试附件
			0	检测到调试附件
3	接收功率	指示源通告的USB Type-C当前功能	高 Z	电源在5V时提供默认或1.5A USB C型电流
			0	电源以5V提供3.0A USB Type-C电流

*

注意：对于值3 -水槽电源：“源供给1.5A / 3.0A USB在5V C型电流源时附接”是基于当电缆连接什么源通告和不指示输出电压是实际上是5V。

- **setPowerAbove5vOnly (uint8_t value)** -取一个整数值 (0或1)。设置为0时，无论协商电压的高低，都会使能输出。设置为1时，仅在连接电源后才启用输出，并为PDO2或PDO3协商电压。
- **setReqSrcCurrent (uint8_t value)** -取一个整数值 (0或1)。设置为0将在RDO消息中请求灌电流作为工作电流。设置为1要求在RDO消息中将源电流作为工作电流。

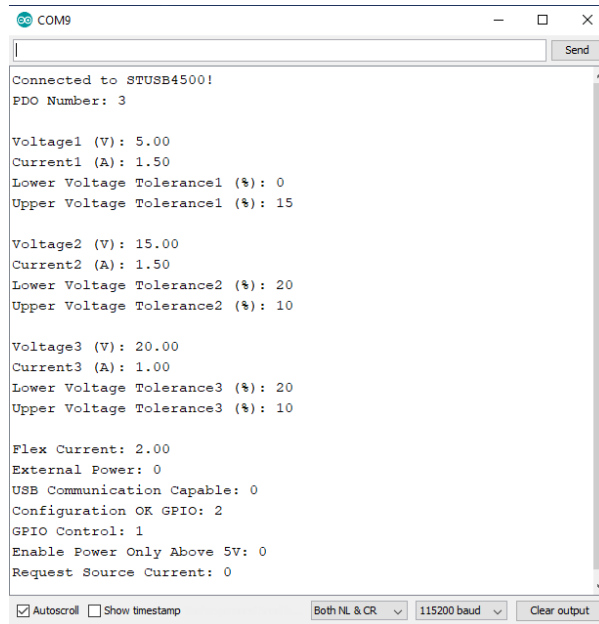
Arduino库示例

在这些示例中，我们将使用Qwiic电缆将配电板连接到RedBoard Qwiic。这些示例中使用的Type-C电源适配器是Apple 87W USB-C电源适配器，其输出为20.3V / 4.3A或14.5V / 2A或9V / 3A或5.2V / 2.4A。

示例1：读取NVM值

在第一个示例中，我们将看到板上当前保存了哪些设置。对于此示例，您无需连接到USB电缆，唯一需要的连接是VDD，GND，SCL和SDA (Qwiic电缆将提供)。

安装库之后，通过单击以下菜单从Arduino菜单访问第一个示例：**文件>示例> SparkFun STUSB4500> Example1-ReadParameters**。选择您的板（在本例中为**Arduino Uno**）和该板枚举到的COM端口。点击上传按钮。打开波特率为**115200**的串行监视器，以查看当前设置。



STUSB4500最多可以存储三个电源数据对象或PDO。每个PDO包含所需的电压和电流以及电压容差。我们可以看到PDO number参数设置为3，这意味着当我们插入电源适配器时，开发板将首先尝试找到与PDO3相匹配的电源选项，该PDO3的电压为20.00V，最大消耗电流1.0A。如果电源适配器能够在20.00V下提供至少1.0A的电流，则合同将被电源接受并切换到更高的电压。如果输出电压超出+ 10%或-20%（16-22V）的容差范围，则宿控制器将与源断开连接并复位。但是，如果PDO3与电源不匹配，它将尝试PDO2的参数，如果仍然不匹配，则电压将保持在5V。

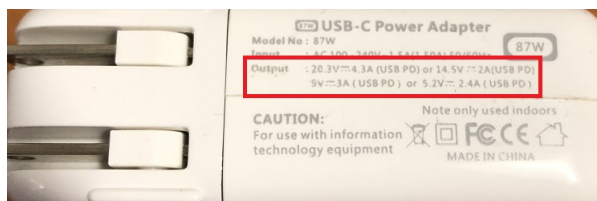
Flex电流设置为2.00A，但是由于没有一个PDO的电流设置为0，因此不会使用该值。我们还看到板子不期望外部电源，也不期望该电源支持USB通信。Configuration OK GPIO设置为2，这意味着在连接源时，如果接受PDO2或PDO3的合同，则对应于该PDO的LED会亮起。GPIO控制设置为1，这意味着GPIO LED当前已配置为错误恢复指示。未设置仅在5V以上的使能电源，因此输出电压将短暂为5V，但在接受合同后将增加。最后，我们看到请求源当前位未设置。

现在我们知道了该板的配置，在下一个示例中，我们将基于电源适配器的功能来配置该板。

示例2：设置NVM值

在此示例中，我们将修改NVM值。下面列出的代码与Arduino中的**SetParameters**示例不同-下面我们设置实际值，而**SetParameters**示例是供将来使用的模板。

在此示例中，我们将需要使用一个电源供应器，该电源供应器是Apple生产的87W USB-C电源适配器，但是还有许多其他适配器可提供一定范围的电压。电源适配器应具有描述输入和输出电源功能的标签，如下所示。



输入电源为100-240 VAC，最大初级电流为1.5A。输出可以提供以下之一：20V / 4.3A或15V / 2A，或9V / 3A或5V / 2.4A。基于这些选项，让我们将电压更改为15V。该电压选项能够吸收高达2A的电流。所请求的电流只能上升到该值才能接受电力输送合同。根据实际输出电压标记了增加的15.3V电压而不是15.0V，但是电源应标出15.0V。该增加是为了补偿在大电流消耗下电缆两端的电压降。我们将要更改的设置是：

- **PDO编号：** 3
- **PDO3电压：** 15.0V

- **PDO3电流:** 2A
- **PDO3上限耐压:** 20%
- **PDO3较低的电压容差:** 20%

将以下代码复制到您的Arduino IDE中，然后将其上传到您的开发板上。上载代码后，打开您的串行端口并确认已应用更改。

```

/*
  Writing New Settings to the STUSB4500 Power Delivery Board
  By: Alex Wende
  SparkFun Electronics
  Date: February 6th, 2020
  License: This code is public domain but you buy me a beer if you use this and we meet some day (Beerware license).
  Feel like supporting our work? Buy a board from SparkFun!
  https://www.sparkfun.com/products/15801

  This example demonstrates how to write new NVM settings to the STUSB4500

  Quick-start:
  - Use a SparkFun RedBoard Qwiic -or- attach the Qwiic Shield to your Arduino/Photon/ESP32 or other
  - Upload example sketch
  - Plug the Power Delivery Board onto the RedBoard/shield
  - Open the serial monitor and set the baud rate to 115200
  - The RedBoard will connect to the Power Delivery Board over I2C write the settings:
    * PDO Number: 3
    * PD03 Voltage: 15.00V
    * PD03 Current: 0.5A
    * PD03 Under Voltage Lock Out: 20%
    * PD03 Over Voltage Lock Out: 20%
  - After the settings are written, the old settings are printed out and then the new settings are printed
*/
// Include the SparkFun STUSB4500 library.
// Click here to get the library: http://librarymanager/All#SparkFun_STUSB4500

#include <Wire.h>
#include <SparkFun_STUSB4500.h>

STUSB4500 usb;

void setup()
{
  Serial.begin(115200);
  Wire.begin(); //Join I2C Bus
  delay(500);

  /* The Power Delivery board uses the default settings with address 0x28 using Wire.

  Optionally, if the address jumpers are modified, or using a different I2C bus,
  these parameters can be changed here. E.g. usb.begin(0x29,Wire1)

  It will return true on success or false on failure to communicate. */
  if(!usb.begin())
  {
    Serial.println("Cannot connect to STUSB4500.");
    Serial.println("Is the board connected? Is the device ID correct?");
    while(1);
  }

  Serial.println("Connected to STUSB4500!");
  delay(100);

```

```
float voltage, current;
byte lowerTolerance, upperTolerance, pdoNumber;

pdoNumber = usb.getPdoNumber();
voltage = usb.getVoltage(3);
current = usb.getCurrent(3);
lowerTolerance = usb.getLowerVoltageLimit(3);
upperTolerance = usb.getUpperVoltageLimit(3);

/* Since we're going to change PDO3, we'll make sure that the
   STUSB4500 tries PDO3 by setting PDO3 to the highest priority. */
usb.setPdoNumber(3);

/* PDO3
   - Voltage 5-20V
   - Current value for PDO3 0-5A, if 0 used, FLEX_I value is used
   - Under Voltage Lock Out (setUnderVoltageLimit) 5-20%
   - Over Voltage Lock Out (setUpperVoltageLimit) 5-20%
*/
usb.setVoltage(3,15.0);
usb.setCurrent(3,0.5);
usb.setLowerVoltageLimit(3,20);
usb.setUpperVoltageLimit(3,20);

/*Write and save settings to STUSB4500*/
usb.write();

/*Read settings saved to STUSB4500*/
usb.read();

Serial.println();

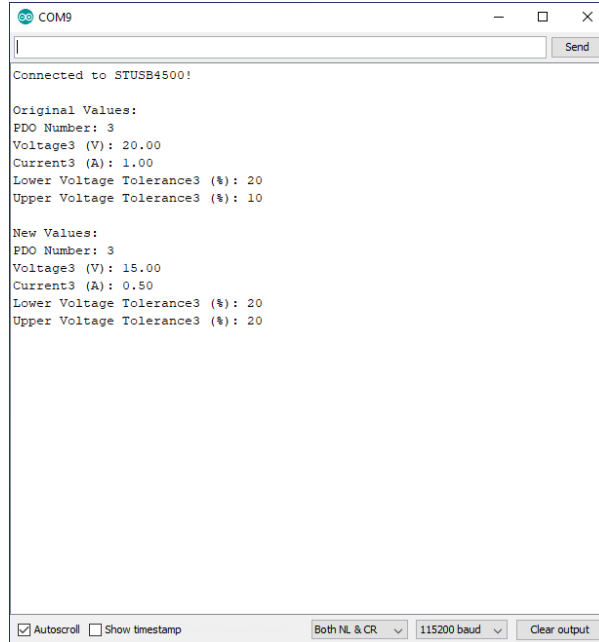
/*Print old setting*/
Serial.println("Original Values:");
Serial.print("PDO Number: ");
Serial.println(pdoNumber);
Serial.print("Voltage3 (V): ");
Serial.println(voltage);
Serial.print("Current3 (A): ");
Serial.println(current);
Serial.print("Lower Voltage Tolerance3 (%): ");
Serial.println(lowerTolerance);
Serial.print("Upper Voltage Tolerance3 (%): ");
Serial.println(upperTolerance);

Serial.println();

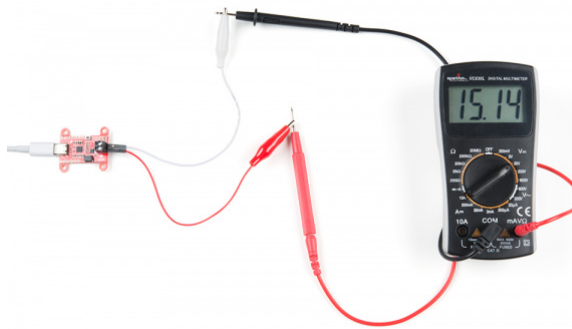
/*Print new settings*/
Serial.println("New Values:");
Serial.print("PDO Number: ");
Serial.println(usb.getPdoNumber());
Serial.print("Voltage3 (V): ");
Serial.println(usb.getVoltage(3));
Serial.print("Current3 (A): ");
Serial.println(usb.getCurrent(3));
Serial.print("Lower Voltage Tolerance3 (%): ");
Serial.println(usb.getLowerVoltageLimit(3));
Serial.print("Upper Voltage Tolerance3 (%): ");
```

```
Serial.println(usb.getUpperVoltageLimit(3));  
}  
  
void loop()  
{  
}
```

上传代码后，以**115200**波特打开串行监视器，并确保已应用设置，如下所示。



重置电源输出板并连接USB-C电缆。如果合同被接受，则PDO3的黄色LED应亮起。用万用表检查电压，如下所示，约为15V。如果LED熄灭，请按复位按钮，并且板应该切换到正确的电压。



单击图像可近距离查看。

故障排除

🔧 需要帮忙?

如果您的产品无法按预期运行，或者您需要技术帮助或信息，请转到SparkFun技术帮助页面以进行一些初步的故障排除。

如果找不到所需的内容，SparkFun论坛是寻找并寻求帮助的好地方。如果这是您第一次访问，则需要创建一个论坛帐户来搜索产品论坛并发布问题。

资源和进一步发展

无法获得我们的电源交付板的足够详细信息？查看以下资源：

- [原理图 \(PDF\)](#)
- [鹰文件 \(ZIP\)](#)
- [GitHub回购](#)
- [STUSB4500数据表](#)
- [Arduino库GitHub存储库](#)

需要灵感吗？查看一些与电源相关的教程：



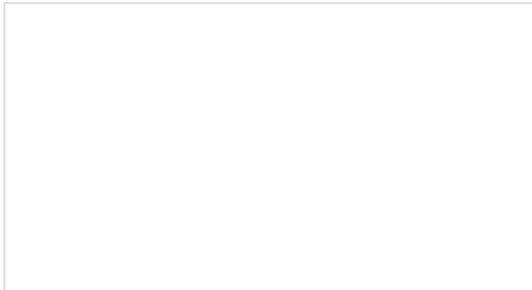
如何构建远程终止开关

了解如何构建无线控制器以在事态发展时断电。



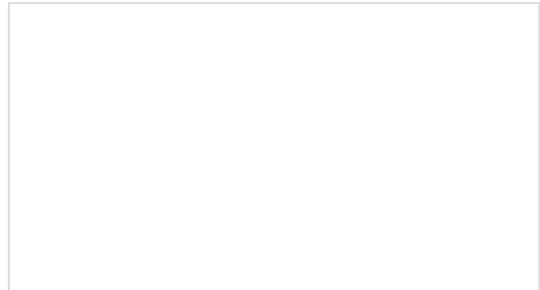
可调式LiPo充电器连接指南

SparkFun可调式LiPo充电器是单节锂聚合物 (LiPo) 和锂离子电池充电器。由于它是可调的，因此该充电器将能够为我们的所有单节电池安全地充电。



LuMini 8x8矩阵连接指南

LuMini 8x8矩阵 (APA102-2020) 是可用的最高分辨率LED矩阵。



初学者零件套件识别指南

初级（甚至是经验丰富的）爱好者的的重要组成部分，为您提供了开始使用嵌入式项目所需的所有基本通孔组件。我们将确定套件中的一些零件，并提供一些基本电路来开始！