

Arduino Shields

This Tutorial is Retired!

Note: This tutorial is for reference only. Most of the shields covered in the "Shieldstravaganza" video series and the updated version of this tutorial, follow the link below.

View the updated tutorial: [Arduino Shields v2](#)

CONTRIBUTORS:  [JIMBLOM](#)

 **FAVORITE** 17    **SHARE**

What is a Shield?

Shields^[1] are modular circuit boards that piggyback onto your Arduino to instill it with extra functionality. Want to connect your Arduino to the Internet and post to Twitter? There's a shield for that. Want to make your Arduino an autonomous rover? There are shields for that. There are dozens (hundreds?) of shields out there, all of which make your Arduino more than just a development board with a blinky LED.



SparkFun catalog manager [RobertC.](#) in awe of an extravagant platter of shields.

Many Arduino shields are stackable. You can connect many shields together to create a "Big Mac" of

Arduino modules. You could, for example, combine an [Arduino Uno](#) with a [Voice Box Shield](#), and a [WiFly Shield](#) to create a [WiFi Talking Stephen Hawking\(TM\)](#).

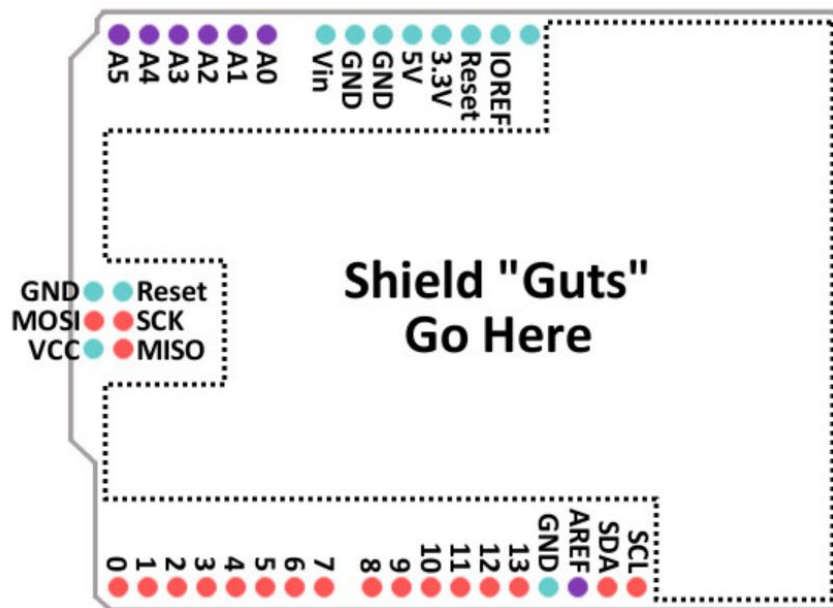


Shields are often supplied with either an example sketch, or a [library](#). So, not only do they just simply plug into your Arduino, but all you need to do to make them work is upload up some example code to the Arduino.

[1] Note: In general, these are called "daughter boards." The terminology and layout depends on the environment platform and form factor. Shields for Arduino typically use the Arduino Uno R3 footprint. However, shields can have a different layout depending on the architecture. Stackable PCBs for the Raspberry Pi are referred to as HATs or pHATs while the BeagleBone refers to them as Capes. We'll focus on the Arduino shields throughout this tutorial.

Shield Form Factor

Every Arduino shield must have the same form-factor as the standard Arduino. Power and ground pins on one eight (previously six) pin header, and analog pins on a six-pin header next to that. Digital pins cover the other edge on the other side, an eight-pin header separated from a 10-pin by that weird 0.5" spacing. Some shields also require a connection to the Arduino's ICSP header (the 2x3 programming header on the end).



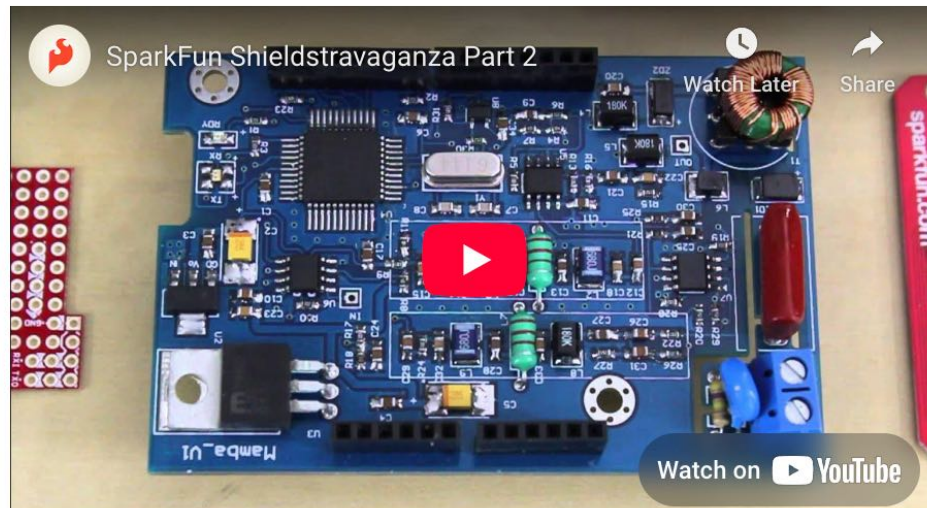
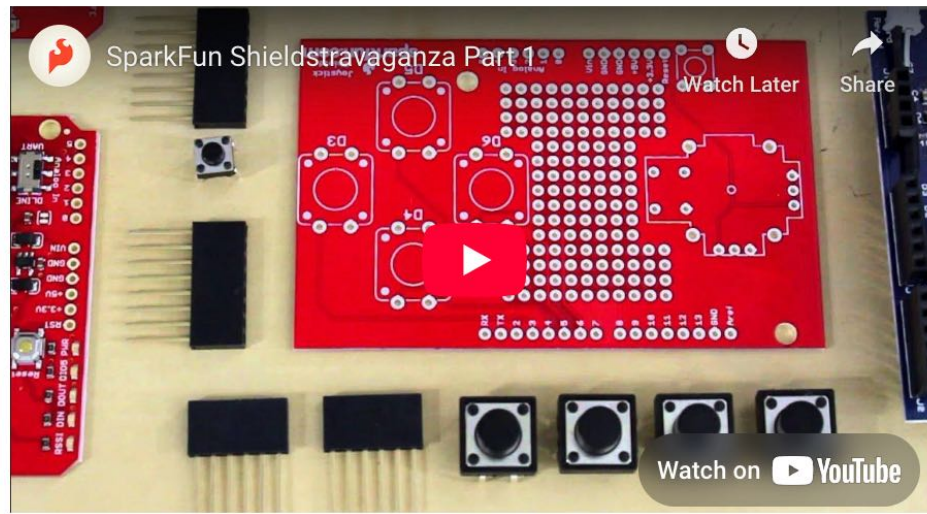
Some shields use every pin on the Arduino, while others only use a couple. When stacking shields, it's important to make sure they don't use overlapping pins. Some shields communicate with the Arduino via [SPI](#), [I²C](#), or [Serial](#), and others use the Arduino's interrupts or [analog inputs](#).

There's a great variety of Arduino shields out there -- too many to ever include in this tutorial. On the next page we'll go over a handful of the more popular and unique shields.

Shieldstravaganza

Here's a list of SparkFun's more popular and unique shields. This isn't an exhaustive list of all Arduino shields (for that, check out shieldlist.org), but it's a good collection. They're sorted into semi-logical categories.

If you're more visually inclined, check out our [ShieldStravaganza Video Series](#) ([Part 1](#), [Part 2](#), and [Part 3](#)). These three exciting videos are filled to the brim with shields, shields, shields, oh...and more shields.

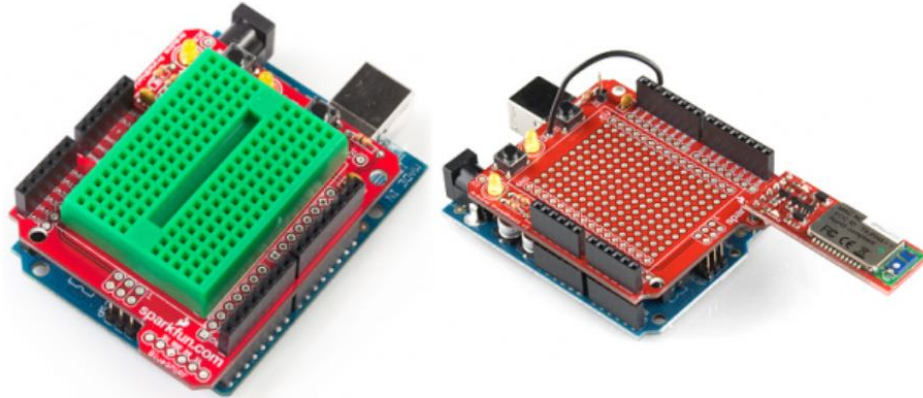


Prototyping (And Then Some)

Prototyping shields don't add much functionality to the Arduino, but they do help in other ways. These shields might do something as simple as breaking out the Arduino pins to screw terminals. In general they

make wiring to the Arduino easier.

- [ProtoShield Kit](#) - The self-titled star of this category. This shield is basically a big prototyping area. You can stick a mini-breadboard on top, or just solder directly to the shield's prototyping area.



- [ProtoScrew Shield](#) - Like the ProtoShield, but each pin is also broken out to a screw terminal. Handy for connecting to external motors or heavy-duty sensors.
- [Go-Between Shield](#) - The intention of this shield is to sit in between two shields. It swaps the pins of the top shield, so they don't interfere with each other.
- [LiPower Shield](#) - This shield allows you to power your Arduino with a [Lithium Polymer](#) battery.
- [Danger Shield](#) - The most awesomest shield ever! This shield is a crazy conglomeration of displays, potentiometers, and other sensors. Great for learning the ins and outs of Arduino or incorporating into audio mixing projects.
- [Joystick Shield Kit](#) - This makes your Arduino a bare-bones controller. With a joystick and four buttons, this makes for a great robot controller.
- [microSD Shield](#) - The Arduino has limited storage space, but this easy-to-use shield (along with the [SD library](#)) allow for plenty of extra storage.

Ethernet, WiFi, Wireless, GPS, Etc.

- [Arduino Ethernet Shield](#) - This is one of the more classic shields. The Ethernet Shield supplies your Arduino with an ability to connect to the world wide web. There's a great [library](#) to support it as well.

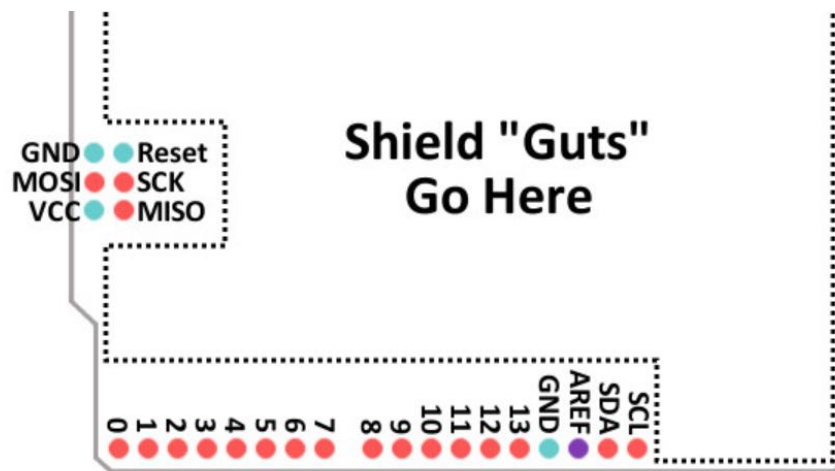


[!] Note: In general, these are called "daughter boards." The terminology and layout depends on the environment platform and form factor. Shields for Arduino typically use the Arduino Uno R3 footprint. However, shields can have a different layout depending on the architecture. Stackable PCBs for the Raspberry Pi are referred to as HATs or pHATs while the BeagleBone refers to them as Capes. We'll focus on the Arduino shields throughout this tutorial.

Shield Form Factor

Every Arduino shield must have the same form-factor as the standard Arduino. Power and ground pins on one eight (previously six) pin header, and analog pins on a six-pin header next to that. Digital pins cover the other edge on the other side, an eight-pin header separated from a 10-pin by that weird 0.5" spacing. Some shields also require a connection to the Arduino's ICSP header (the 2x3 programming header on the end).





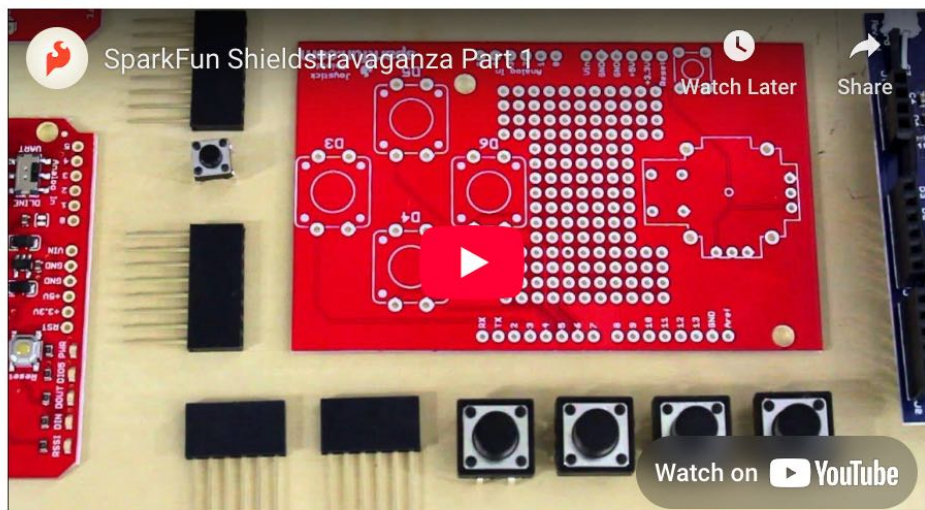
Some shields use every pin on the Arduino, while others only use a couple. When stacking shields, it's important to make sure they don't use overlapping pins. Some shields communicate with the Arduino via [SPI](#), [I²C](#), or [Serial](#), and others use the Arduino's interrupts or [analog inputs](#).

There's a great variety of Arduino shields out there -- too many to ever include in this tutorial. On the next page we'll go over a handful of the more popular and unique shields.

Shieldstravaganza

Here's a list of SparkFun's more popular and unique shields. This isn't an exhaustive list of all Arduino shields (for that, check out [shieldlist.org](#)), but it's a good collection. They're sorted into semi-logical categories.

If you're more visually inclined, check out our [ShieldStravaganza Video Series](#) ([Part 1](#), [Part 2](#), and [Part 3](#)). These three exciting videos are filled to the brim with shields, shields, shields, oh...and more shields.

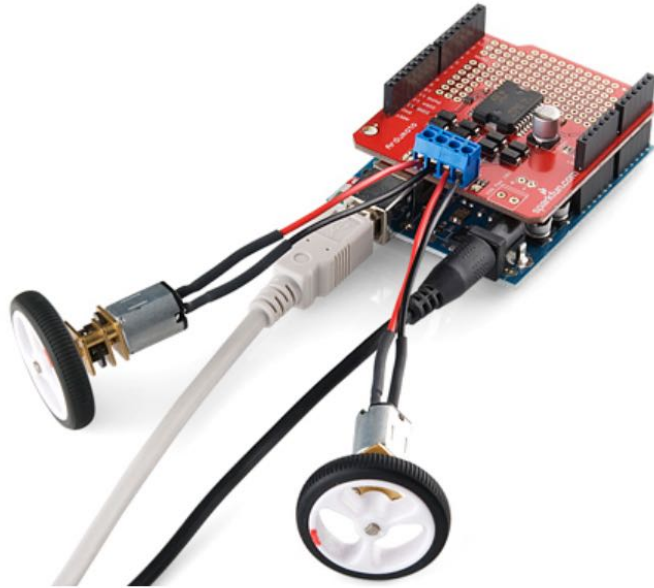


- [EL Escudo](#) - Electroluminescent wire is awesome! Use this shield to add up to eight strands of EL wire to your project. You can finally make that Arduino-powered Tron costume.

- [CMUcam](#) - This camera module adds vision to your Arduino. You can use it to track blobs, so your robot doesn't hit any traffic cones.

Motor Drivers

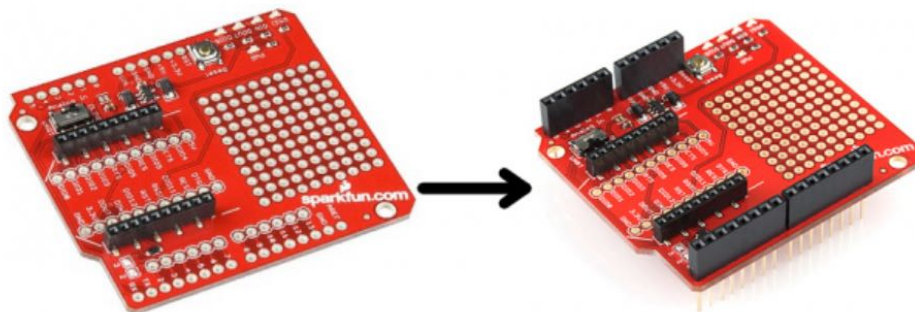
- [Ardumoto Motor Driver Shield](#) - This classic motor driver shield can control two DC motors.



- [Monster Moto Shield](#) - If you need to drive beefier motors than the Ardumoto Shield can handle, this is the next step up.
- [PWM Shield](#) - Usually when you think [pulse-width modulation \(PWM\)](#), you might think "dimming LEDs", but PWM is also used to drive servo motors. This shield can be used to drive your crazy 12-servo hexapod.

Installing Headers (Preparation)

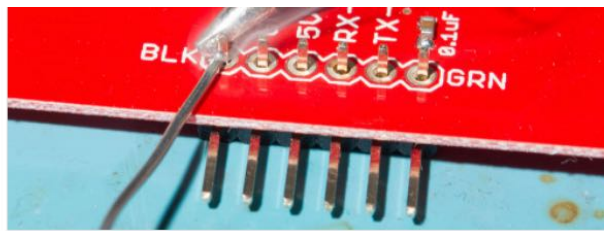
Many shields come without any headers attached. This keeps their final fate open to your interpretation (maybe you'd rather use straight male headers, instead of the usual stackable headers). The following pages will explain how you can turn your drab, header-less shield into a fully functional, ready-to-plug-in module.



Required Tools and Materials

Shield assembly requires soldering. Solder helps make a good, physical and electrical connection. Without solder, connection between shield and Arduino will be intermittent (at best). If this is your first time soldering, consider checking out our [how to solder tutorial](#).





How to Solder Through-Hole Soldering

SEPTEMBER 19, 2013

This tutorial covers everything you need to know about through-hole soldering.

You'll need these parts in order to install headers on your shield:

- **An Arduino shield** - Any shield should do. All Arduino shields should have a standard Arduino footprint.
- **4 headers** - The amount of pins on the headers depends on whether your shield has the newer *R3* footprint, or the original Arduino layout.
 - Original: (2) 6-pin and (2) 8-pin headers
 - R3: (1) 6-pin, (2) 8-pin, and (1) 10-pin header

And these tools are required you'll need:

- **Soldering iron** - The most basic soldering iron should work (flavors include [US](#) or [Europe](#)).
- **Solder** - If you value your health, use [unleaded solder](#). If you value your time, use [leaded solder](#).
- **A wet sponge** - This'll keep your iron tip clean and shiny. Any wet sponge will do. Use the one included with the iron stand, or get a fancy [brass sponge](#).

These tools are completely optional, but may make your life a bit easier:

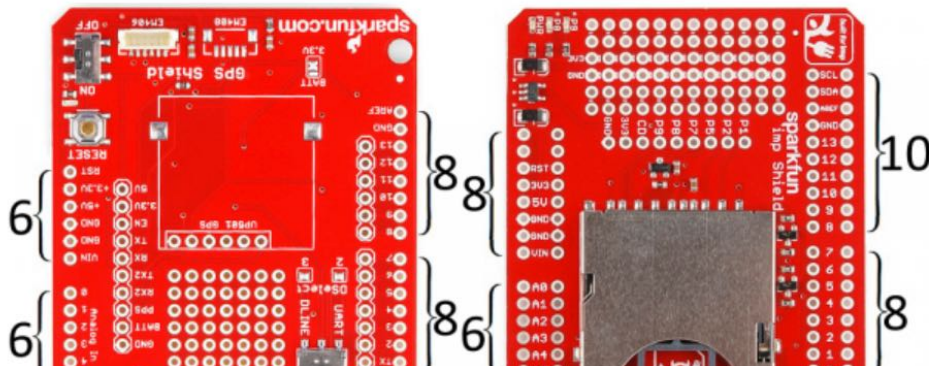
- [Soldering iron stand](#) - This'll keep your soldering iron off your floor and out of your lap (ouch!).
- [Third hand](#) - If you're running out of hands and can't get a *trusting* family member to hold something for you, this'll do the job.
- [Solder wick](#) may come in handy if you need to remove solder from a joint.

Preparation

Before you start warming up that soldering iron, let's take some time to plan out the assembly process.

DO YOUR HEADERS MATCH YOUR ARDUINO SHIELD?

From the inception of Arduino until some time in 2012, every Arduino had the same standard footprint: two 6-pin headers on one side, two 8-pin headers on the other. Lately, though, Arduinos have been transitioning to a new shield-header layout, called the **R3 footprint**. This layout has a 6-pin and 8-pin headers on one side, and an 8-pin and 10-pin on the other.





Old Style

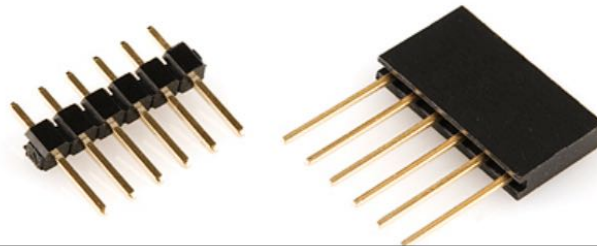


New "R3" Style

Make sure the headers you have match the pin-out of your shield! Also consider if your Arduino layout matches your shield's. **R3 Arduinos should be reverse-compatible** with shields of the older footprint, however, older Arduinos are not completely compatible with new shields of the R3 footprint (something about sticking 10 pins into an 8-pin header).

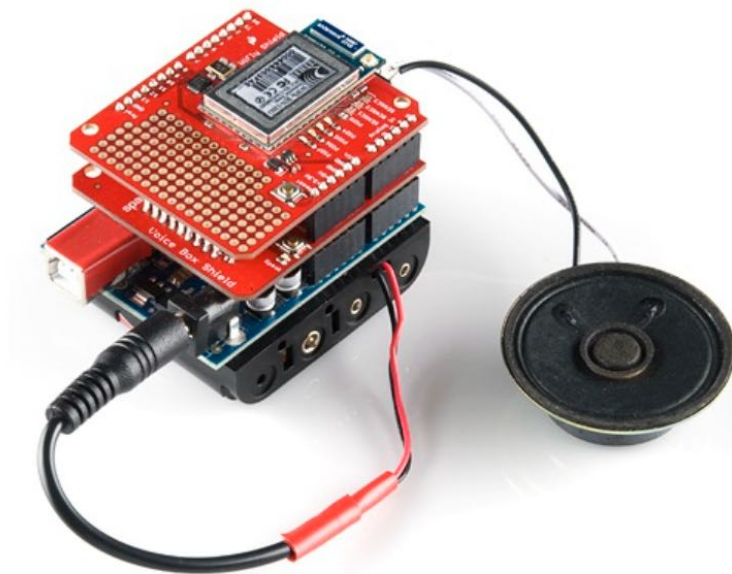
What kind of header should you use?

There are all kinds of headers, but there are only two that are recommended for installing on shields: stackable or male.



A straight male header (left) and a stackable header (right).

Stackable headers are especially great for stacking shields. They also maintain your ability to plug a jumper wire into any of the Arduino's pins. **This tutorial will explain how to install stackable headers.** Stackable headers are available in [6-pin](#), [8-pin](#), and [10-pin](#) varieties, or you can buy the headers in packs for [original](#) or [R3-type](#) shields.



The glory of stackable headers. They enable you to make a [wireless talking Arduino](#). Notice the top shield has male headers, the bottom shield has stackable.

Simple, straight [male headers](#) are also an option for connecting a shield to an Arduino. Male headers are beneficial in that they create a **lower-profile stack** when connected to an Arduino. If you plan to stuff your Arduino/shield combo into an enclosure, you may need to consider using male headers. This tutorial focuses on stackable header installation, check the [Tips and Tricks](#) section for male header assembly instructions.

Do not install female headers, right-angle male headers, swiss machine-pinned headers, round headers, or

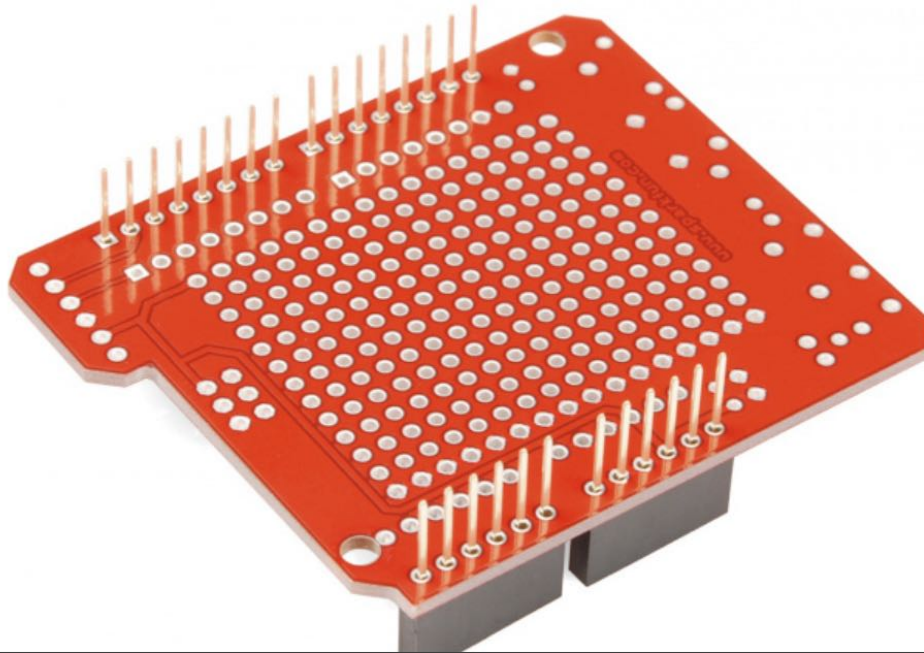
a variety of other headers that may exist out there. You really should only use headers that have straight, rectangular, male pins.

Now then, plug in and start warming up those soldering irons. It's time to get soldering!

Installing Headers (Assembly)

Step 1: Insert All Four Headers

Plug all four headers into the shield. Make sure you **insert them in the correct direction**. The male pins of the header should enter the top side of the shield and extend out the bottom. This orientation is of utmost importance. Don't solder anything until you've got the headers going the right way!



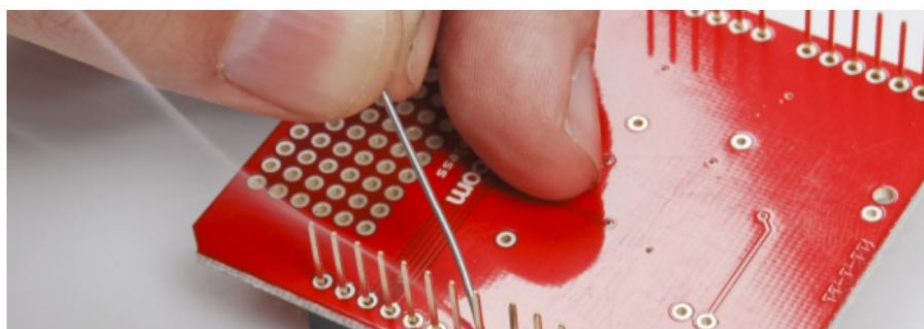
Headers inserted, aligned as well as possible, ready to solder.

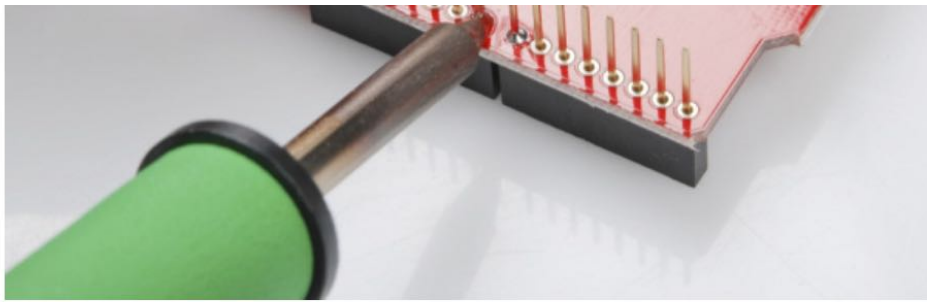
With the headers inserted, flip the shield on its top-side to rest on the black, female side of the headers. Hopefully you've got a nice flat workspace to lay it on. Try to align all headers so they're **precisely perpendicular** to the shield PCB.

Step 2: Solder One Pin On Each Header

Finally, soldering time! It's important that each of the headers is at a nice, 90° angle to the PCB. This will ensure that the shield slides straight onto your Arduino, and you won't have to bend any pins in doing so.

In order to guarantee that each header is straight, **start by soldering just a single pin on each**. If they're at a weird angle, it'll be much easier to re-heat just a single pin, while adjusting the alignment.



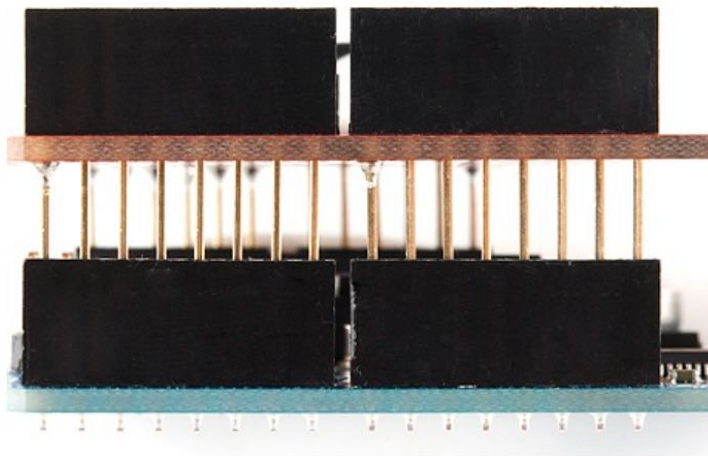


One pin down, one in progress, two to go. Soldering one pin on each header.

Four solder joints down, only 24 (to 28) to go!

Step 3: Check Header Alignment

With those four pins soldered, try plugging the shield into your Arduino to test the header's alignment. Make sure your Arduino's not powered while you do this alignment check.

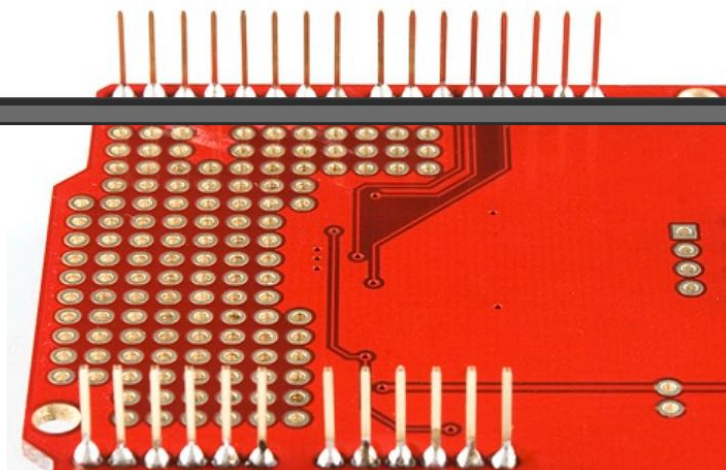


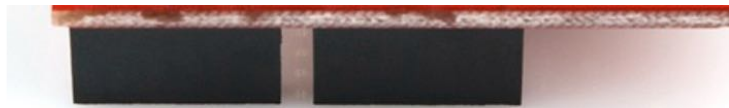
Temporarily plugging the shield in to check that all pins line up.

Does everything line up? No pins bending? If not, find the guilty header and try to re-align it. Warm the joint back up with your iron, and slightly move and adjust the header alignment. Also, be careful when pulling the partially soldered shield out of the Arduino. Since all the headers are not soldered, you could easily bend them as you pull it out of the Arduino's female headers.

Step 4: Solder All Remaining Pins

If your headers are all lined up, you can attack the remaining unsoldered header pins. When you're done, you should have 28 (or 32) shiny volcanoes of solder.

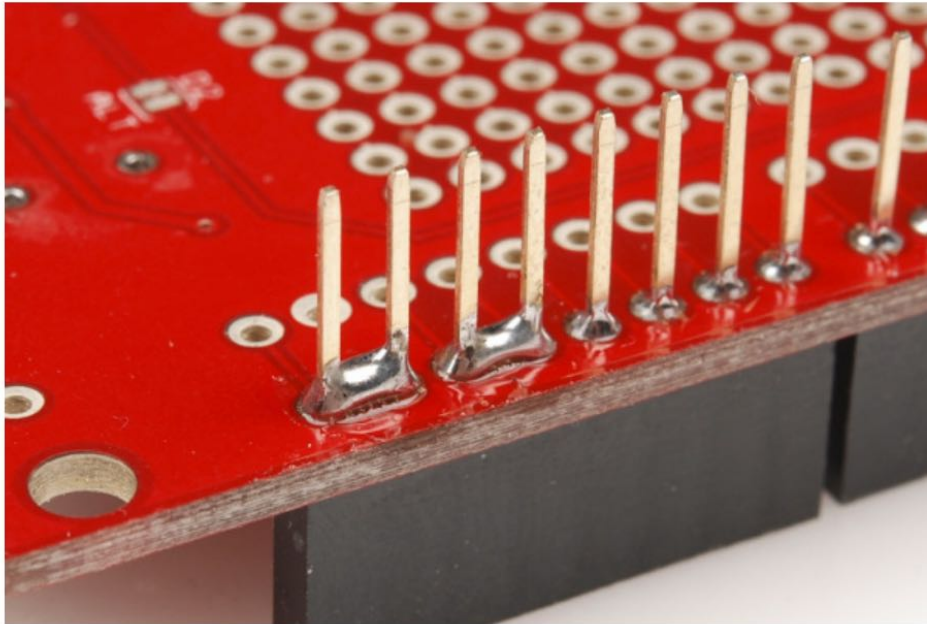




That's a beautiful sight. Everything's soldered up.

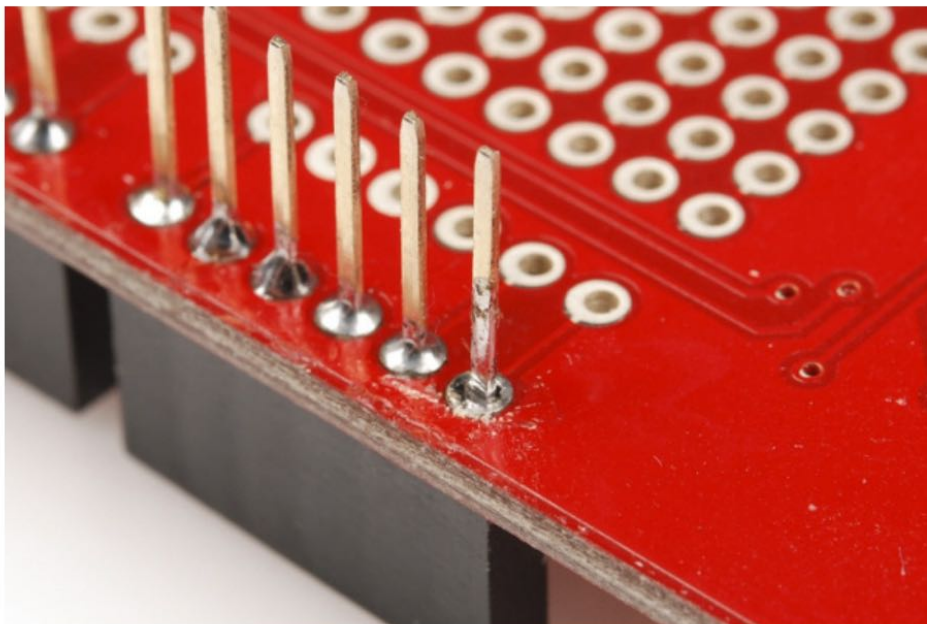
Step 5: Check For Shorts or Cold Joints

With everything soldered, double check for bad solder joints. Did any of your joints stray into another creating a short? If so, you can take some [solder wick](#) to the joint, or just try re-heating the short and "pushing" the solder where you want it.



Well, that's just egregious! Watch out for shorted solder joints like that.

Also check for **cold solder joints** - a joint that's got some solder on it, but isn't quite connecting the two solder points together. Cold joints aren't always the easiest to see; look out for joints that aren't as shiny, or pins that still seem loose.

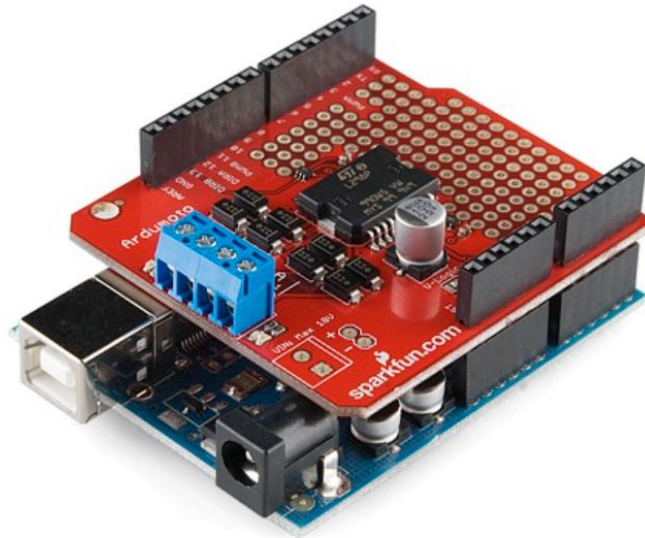


That last pin could use a bit more solder. It doesn't quite look like a connection's been made.

To fix a cold joint, re-heat the solder on the pin, and add just a bit more.

Step 6: Plug It In!

It's usually best practice to power down (unplug) your Arduino before you connect a shield to it. Hopefully all of the pins are still well-aligned and the shield just slides right into the Arduino. Take care not to bend any pins while inserting, and make sure they all go into their mating female headers.



That satisfying feeling as the shield slides straight into your Arduino

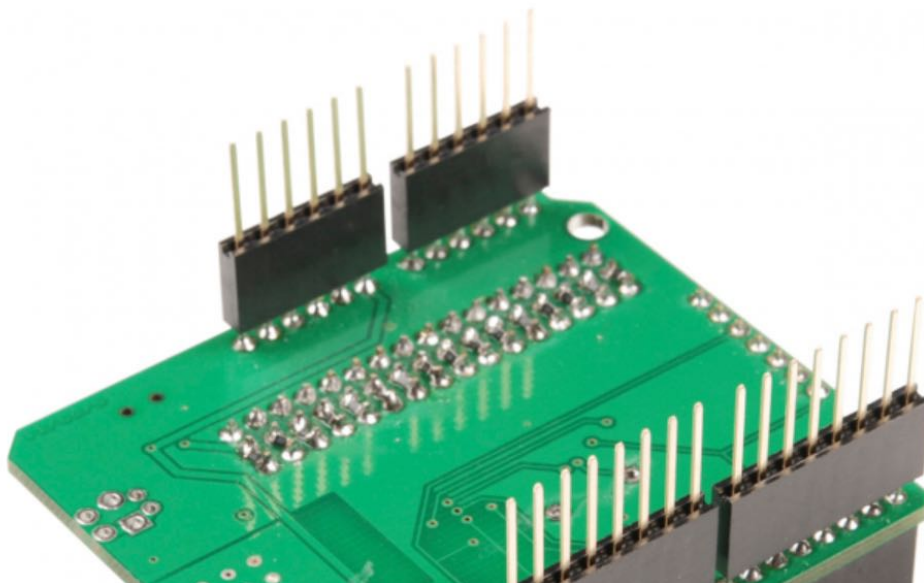
Assembly Tricks

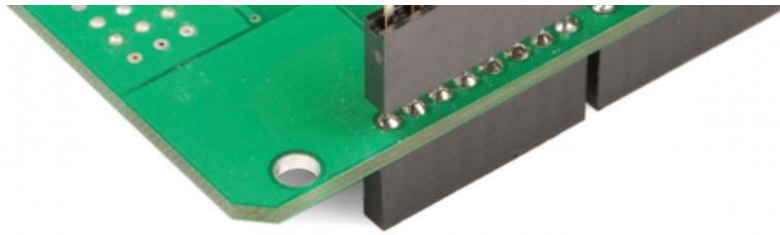
The previous assembly page should detail everything you need to know about simple shield header installation. There are, however, a few tricks we've picked up along the way...

Use An Old Shield to Aid Alignment

The easiest place to mess up shield assembly is in aligning each of those headers. It's best to avoid soldering the stackable headers while the shield's connected to the Arduino, so the method described in the assembly section is usually best. If you've got a spare shield lying around, you can take advantage of another little trick by using it as a header-alignment-jig.

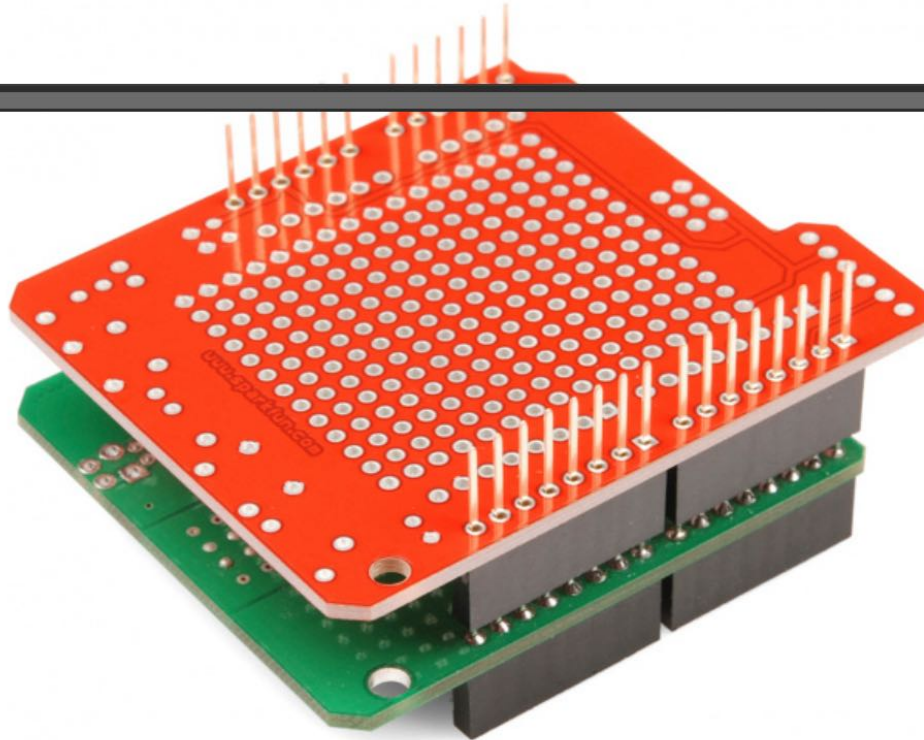
Begin by plugging all of the headers into your spare shield jig.





The green shield will be used as our jig. First, insert the stackable headers into it.

Then insert the headers into your to-be-soldered shield, and solder them all up. Assuming the spare shield is well-aligned (you may want to check that first), it should handle all of the alignment of your new headers.

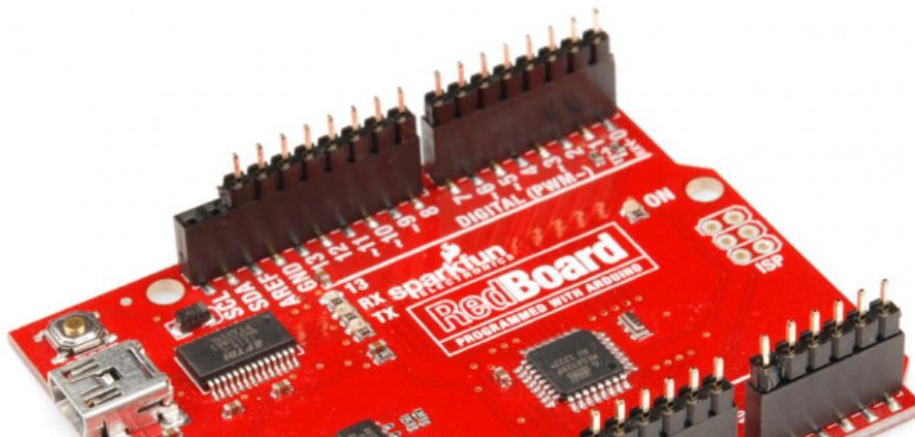


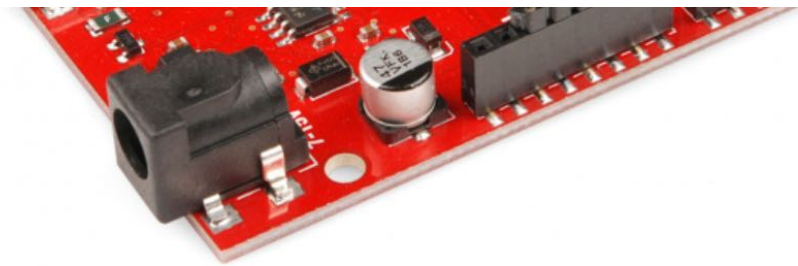
The jig should correctly align all of the headers. Solder away!

Installing Male Headers

If you value a smaller profile shield installation over the ability to stack shields and connect jumper wires, male headers are an option.

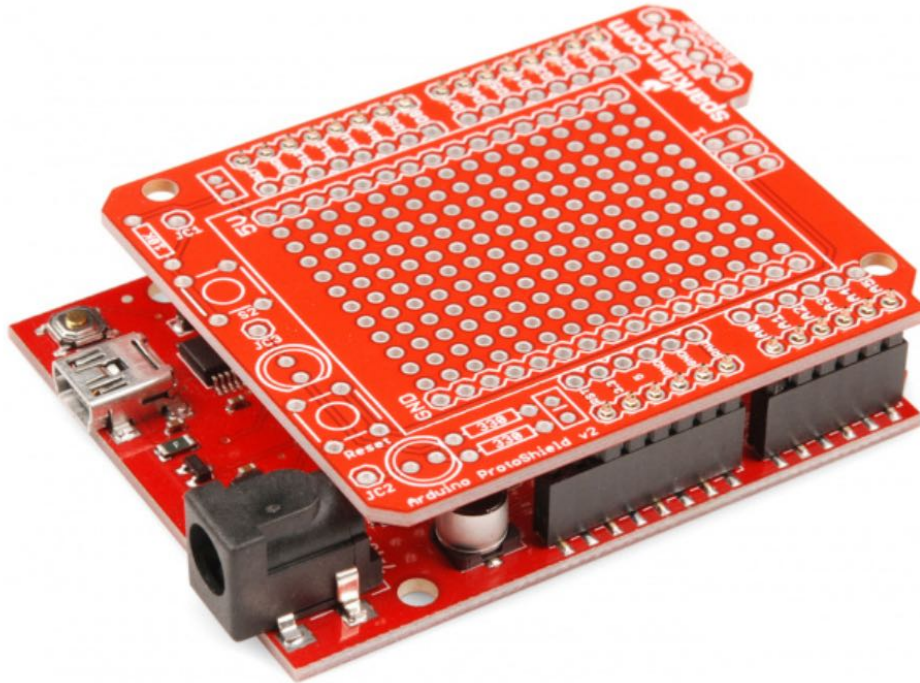
In a way, male headers are actually easier to align and install, because you can use your Arduino as a jig. Begin by inserting the headers into your Arduino.





RedBoard's make an especially special jig for aligning male headers.

Then line up and plug in the shield, and solder away.



Shield with headers, ready for soldering. We can trust the Arduino to line the male headers up for us.

Be somewhat careful using this method, don't leave the iron on the pins for too long or you risk burning the Arduino's headers. If you're especially worried about burning your Arduino's female headers, you can solder just a single pin on each header, remove the shield, and solder the rest.