# Really Bare Bones Board (Arduino ) Assembly Instructions

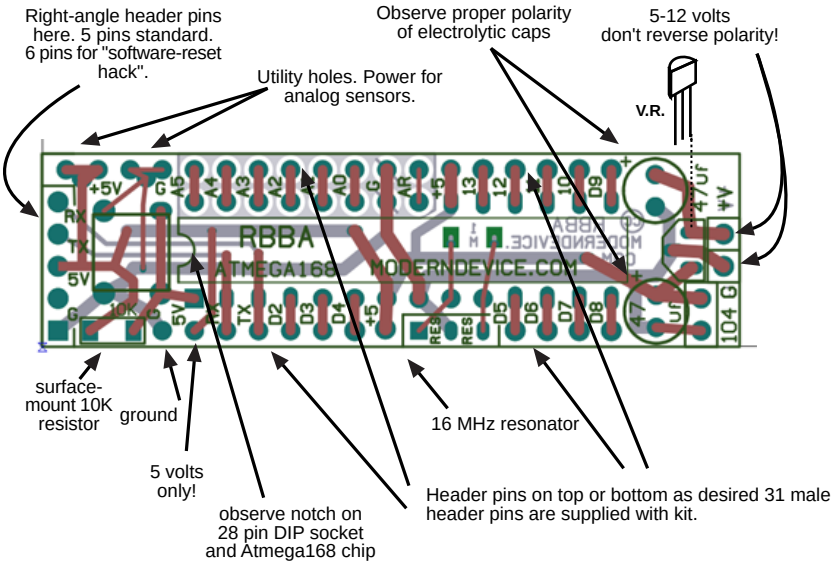



**figure 1: BBB layout**

## Kit Parts List

**S.M. Resistor**

1 10k (marked 103)
1 1k resistor for LED (optional, not mounted on RBBB)

**Capacitors**

1 .1ufd (104) ceramic, 25 V
2 4.7-47 ufd tantalum or electrolytic capacitors (polarized), 25V

**Semiconductors**

1 L4931CZ50LDO 5V, low-dropout voltage regulator, TO-92 package
**optional** - LM2937 500 mA LDO regulator TO-220 package
1 3mm LED (optional, not mounted on RBBB)
1 Atmega168 28 pin DIP package preprogrammed with bootloader
1 16 MHZ ceramic resonator with built-in capacitors, three-terminal SIP package

**Hardware**

31 snap-off .100" center male header pins , or .100" female headers as desired.
6 right-angle male-header pins .100" centers (see text)
1 momentary switch
1 28 pin (narrow .3") IC socket OR 2, 14 pin DIP sockets

**Programming Connection**

FTDI TTL-232R USB programming cable FTDIchip.com
or Modern Device P3/4 serial port programming adapter

**Auxillary Power**

5 Volt - 1A power adapter (optional)
All Electronics CAT# PS-504
9 volt battery

The Really Bare-Bones Board has been engineered to be the smallest, easy-to-construct Arduino-compatible, specially aimed at students and prototypers. The board plugs into a breadboard, or can be embedded in small or large projects such as toys, small appliances and wearable-computing projects.

The Really Bare-Bones Board includes all of the functionality of all other Arduino boards, except for the physical limitations of its form factor..

The boards can be built in a half hour by a beginner at soldering, or in 20 minutes or less by someone who has built a few of them before.

Only a low-wattage soldering iron (15 - 25 Watts), and solder is required.

### Assembly Instructions

Start assembling the board with the smallest, lowest profile components first. Also when parts are inserted, and the board is flipped over to solder, components will tend to stay in the board better, if similar, low-profile components are inserted first.

For beginners that have never done electrical soldering before see the Soldering section before starting. There are a few tips for multiple-unit "power builders" too.

• Solder in the surface mount resistor first. Align the resistor on the pads and just touch one pad (not the resistor) with soldering iron and solder. The solder will wick under the resistor almost instantly. If it isn't straight, reheat and straighten it before similarly touching the other end with soldering iron and solder.

diodes, small ceramic capacitors, LED, reset switch, and voltage regulator next.

• The resistors and smaller capacitors have no polarity and may be inserted in either direction.

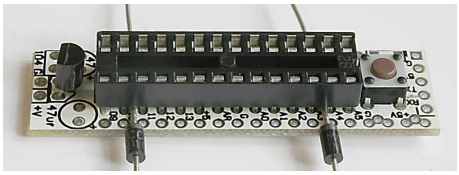
• Make certain that polarized components, such as diodes,voltage regulator, and larger electrolytic caps are inserted in the correct direction.

• Mount the two 47 uF electrolytic capacitors, carefully observing the polarities and orientation of each. The long leads on the capacitors are positive.

• Insert the IC socket (not the chip) into the board, taking care to align the notch with the notch on the board silkscreen. You may wish to use some resistor or diode leads to space the socket off the board slightly, as shown in the photo. This makes the pin labels visible underneath the socket.

*For larger and multi-pin components, such as the chip socket and header pins, there is a little trick that may be helpful to get them mounted neatly.*
*Solder in one pin only, or in the case of the socket, two*

*diagonal corner pins. Then flip the board over to inspect it. If the component is not mounted tightly down on the board, simply put a little pressure on the component with your index finger while reheating the soldered pad(s) with the soldering iron, this will get the part mounted down flush before you solder in the other pins.*



**Spacing the socket off the board with some resistor or diode leads will make the pcb labels more legible**

- Solder in the header pins. The board is made to be easy to customize for particular applications. The following instructions are for the most standard orientation for header pins, but feel free to mount (or omit them) as you wish. In some installations it may be more robust and reliable to solder wires directly to the board.
- Mount the resonator
- For the side of the board with the resonator, snap off an eight pin section of header pins and a five pin section, mount them on the bottom of the board and solder them on the top.
- For the far side of the board with the resonator, snap off a 14 pin section of header pins mount them on the bottom of the board and solder them on the top.
- The extra ground and +5V holes on the left end of the board are made to be utility holes for connecting sensors and the like, put pins in them or leave the empty as you wish.
- Note that an LED and a 1k resistor is provided for a pilot light but there is no place on the board for it. This was by design as it was desired to make the board as small as possible to hide in toys, wearables and generally go into tight places. The suggested place for the pilot is on pin 13 as download status blinks and bootloader reset blinking occurs on that pin.See the breadboard drawing.

**Software-reset hack. README FIRST**

Before adding the switch and programming header take a minute to consider your options. Hopefully this won't sound impossibly complicated because it's really not, but we have to give you some background on the bootloader to make this clear.

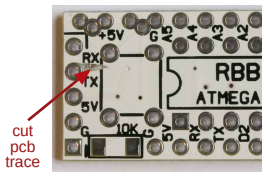
In the Arduino 0009 IDE (and later versions) there is an option to send a reset-signal to the Arduino, on models that have a capacitor between the RTS/DTR line and the reset. The RBBB board was designed just before the Arduino people made this new feature public, so the RBBB does not have the new capacitor - and the board was even designed to eliminate the RTS pin (to save space).

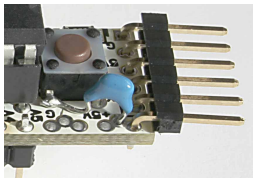
In any case the hack is pretty simple to add the pin and capacitor if you want to do it. There are photo's below to show you what's involved.

Here's our analysis: If your RBBB is destined to be inserted into a tight space such as toy or a prototype where the reset switch is inaccessible, then the auto-reset seems like a good idea and you should probably do it. If your RBBB is anywhere that the reset switch is accessible, then the hack is probably not worth the fuss.

So the question is, what does hacking in this capacitor buy me? That requires a bit of explanation. The Arduino team designed a new bootloader for the Diecimila, which is the first "official" Arduino board that has the auto-reset capacitor on it. The new Diecimila bootloader has some nice features, but it is incompatible with other equipment that does not have the capacitor installed. Consequently Modern Device, Wulfden.org and others are using another bootloader, "Adaboot", which is a hack of both the "official" bootloaders. It combines the best features of both bootloaders and can be used on any hardware. The RBBB is shipped with Adaboot. You can read more about Adaboot on the M.D. website. One of the great things about Adaboot is that it does away with almost all of the annoying waits that were built into the older "NG" bootloader.

To sum things up, this removes much of the incentive to do the hack. So we recommend that users considering applications where the reset switch is accessible just build the kit as designed, and use the reset switch to upload sketches to the board.



* Note that a previous version of this documentation showed the cap connected to the wrong lead on the switch the switch.

**Software-reset hack.**

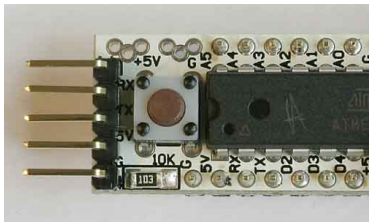
- 1) Cut the trace between +5V pad on the programming header and the pads at the top of the board. You will have to look closely to see it. An Exacto knife flipped over on the "dull" side and pushed, works well for this. This it with a multimeter to make sure it's really cut.

2) Solder in the surface mount resistor first, and the six pin programming header. It will be slightly crooked because of the "+5V" hole is out of line with the rest of the header pads.

3) Solder in a .1 ufd (104) cap as shown, between one of the (former) +5V holes and the nearest corner of the reset switch. That's it, you're done.

4) If you are programming the RBBB (or BBB) with Windows, you'll need to adjust the following serial port settings:

Device Manager -Comm Ports - USB Serial Port - Port Settings - Advanced button - Set RTS On Close



### Standard construction

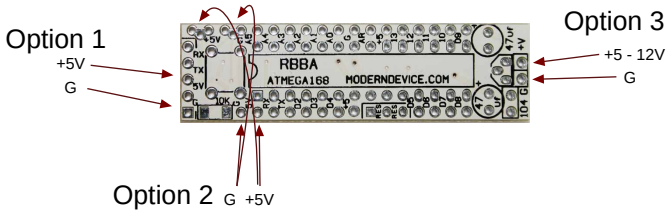
- Here's the standard construction, for comparison. Note that the chip is soldered in directly in this unit, instead of using a socket - this aspect is non-standard and sockets are included in the kits.

### Clean Your Board

- Hobbyists and prototypers often omit this step, but it's a good idea to clean the solder flux off your board with a toothbrush and isopropyl (not denatured) alcohol. Scrub and rinse with clean alcohol until the board looks flux-free. Dry it off a little with a paper towel or rag and have a good inspection of your solder joints, to make sure pins are not bridged (shorted) with solder, and that all the solder joints look shiny, smooth, and cover the pads completely. Questionable solder joints may be fixed simply by reheating with the soldering iron.

### Testing your RBBB Arduino

- Hook up a LED and a 1K series resistor on pin 13. Hook up an FTDI cable (or P3/P4 programming adapter), or apply 5-12 Volts and ground to the +V connection near the voltage regulator. Press the reset button to see if the LED flashes 4 times, followed by another flash. If this doesn't happen, unplug things right away and see the troubleshooting section.
- Boot up the Arduino application and try downloading the blinking light program. Push the reset button on the board and click the download icon at about the same time. If you did the capacitor hack, don't press the reset button.
- Recently shipping versions of Adaboot should show four flashes a pause and another flash on reset, either done manually, or from software reset.
- If your board doesn't seem to work, see the troubleshooting guide on page 6.




**Powering options for the RBBB**

### Powering the RBBB Arduino.

Because the RBBB was designed to be small and simple, the power system is also simple, but there are some things to watch for in powering your RBBB.

Option 1) USB cable.

This will power your RBBB off a laptop or desktop through the FTDI RS232 to TTL cable.

Option 2) If you have some kind of 5V **regulated** power on your breadboard, you can power the RBBB from the 5 volt rails. These pads and the ones on the top left of the board are connected to the output of the voltage regulator.

Option 3) 5 to 12 volts may be supplied to these pads, which connect to the input of the voltage regulator. A convenient battery powered option would be a 9 volt battery connected to these inputs.

**Caveat:** You should not hook up two power sources at the same time. This means, for example that you should leave the external power disconnected while programming the RBBB. Another option might be to just cut off the +5V pin of the programming header so that power cannot be supplied from the cable. We realize this is slightly inconvenient and will put some header pins for a power switch (similar to the BBB) on the next board revision.

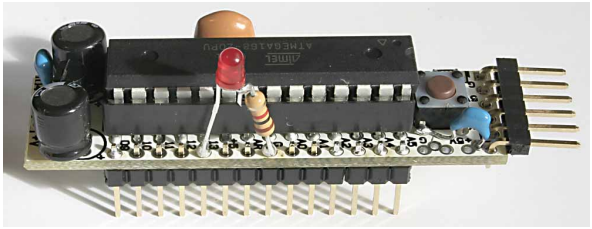
**Further caveat:** Normally we like to provide backwards diodes across power inputs to protect the circuit from reverse power connections. Because the design goal was to keep the RBBB as small as possible, we didn't include one on this board (but may in future revisions). (You could solder one in yourself in two of the power "utility pads" on the top left of the board.)

Consequently, if power is applied with reverse polarity, you may well kill the Atmega chip, (but probably not the voltage regulator, which is made to withstand adversity). To avoid this, check power connections twice before applying power, and disconnect the power supply right away if the board doesn't blink when it is supposed to.

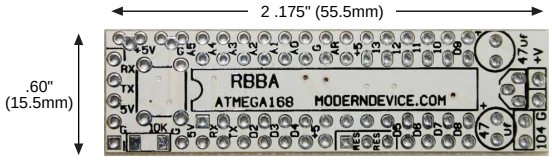
### Pilot Lights

A pilot light and resistor is supplied with your kit. There are no specific pads for it on the board. One option might be to solder it in permanently to pin 13 and ground. A pin 13 LED has the advantage of showing status flashes (reset and upload) that appear on the pin.

Another option might be to put it in across the power supply lines, say at one of the "Option 2" locations. A pilot light across the power supply lines has the advantage of showing instantly if the power is hooked up correctly. Feel free to use both, especially if you aren't using battery power.



**One possible pilot light scheme. The pilot light could also go on a breadboard, or get soldered into the +5V and ground utility holes. (If the auto-reset hack is not performed.)**

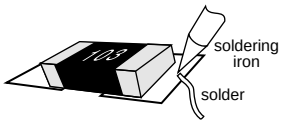



**The Real Bare-Bones Arduino board shown actual size**

## Surface-Mount resistor

schematic symbol

10k (marked 103)



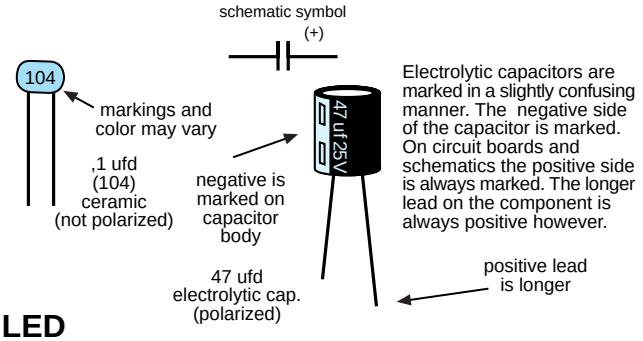


Place the surface mount resistor on the pad, poke it into place with an Exacto knife or spare resistor wire. Touch the soldering iron and solder to the pad, unlike other soldering, do not touch the resistor with the soldering iron. Solder will rapidly flow onto the pad and resistor, lift the soldering iron immediately. If it ends up extremely crooked or up in the air, poke it down with an Exacto knife or wire while reheating the pad. Get it fairly well aligned before soldering in the other side of the resistor.

Don't worry if the resistor is not on perfectly straight, the electricity can't tell the difference and it's too small for your friends to see.

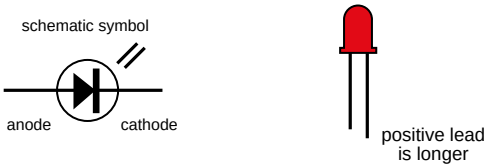
## Capacitors

Capacitors are components that store electrical energy (charge). There are several different technologies that are used to construct capacitors. The BBB contains ceramic capacitors, that are not polarized, and electrolytic capacitors, that are polarized, and must be inserted with the correct orientation.



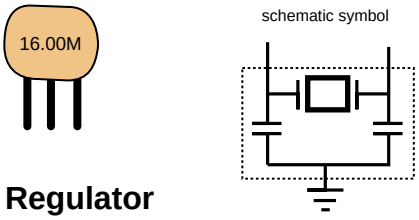

## LED

LED's are diodes which emit light. They are polarized like diodes so insert them in the correct direction.




## Resonator

The 16 Mhz ceramic resonator acts as a calibrated oscillator for the BBB. As you can see from the schematic, it contains a crystal element and two small capacitors. It's symmetrical, so you can't put it in backwards, but it is a good idea to put it in so that you can read the label.

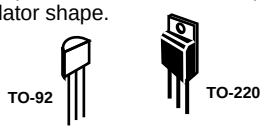


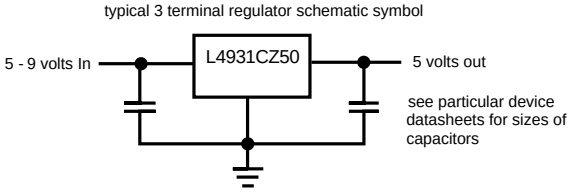

## Voltage Regulator

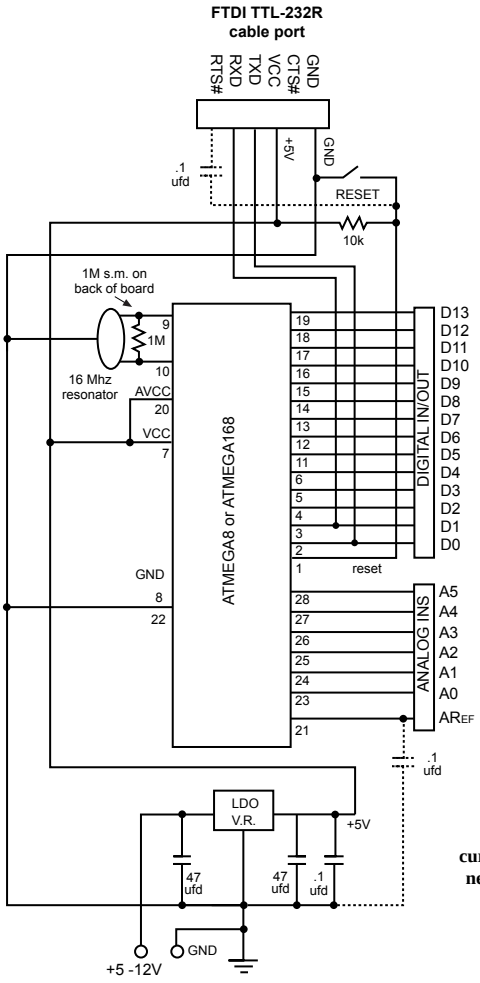
The voltage regulator is an integrated circuit which will limit higher input voltages to 5 volts. It will also limit the current flow in case of short circuits. The electronics industry calls the physical form an IC is packaged in a "package" or "case", the actual IC is always a small chip embedded somewhere in the plastic. The voltage regulator provided may vary by number but is in a TO-92 case.

If you need to have more regulated power, to power a lot of high powered LED's on a breadboard, for example, then, your board will accommodate a TO-220 package regulator, such as the LM2937 listed in the parts list, or the popular 7805. Just remember to insert it backwards as shown in figure 1.

Make sure you get the TO-92 regulator inserted in the correct orientation. It is **not** symmetrical so match the part outline on the board with the regulator shape.



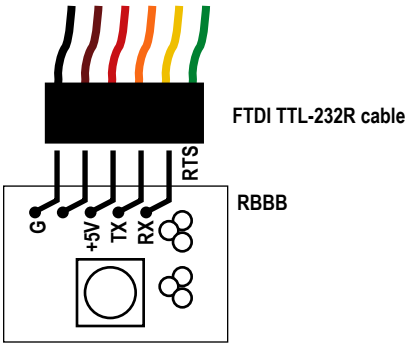






**Real Bare-Bones Board / Board Schematic**
**RBBB**
**Arduino® - Compatible**

**RBBA/B Schematic**
**Unlike some other Modern Device products, this schematic and board design are in the Public Domain.**
**Paul Badger 10/2007**

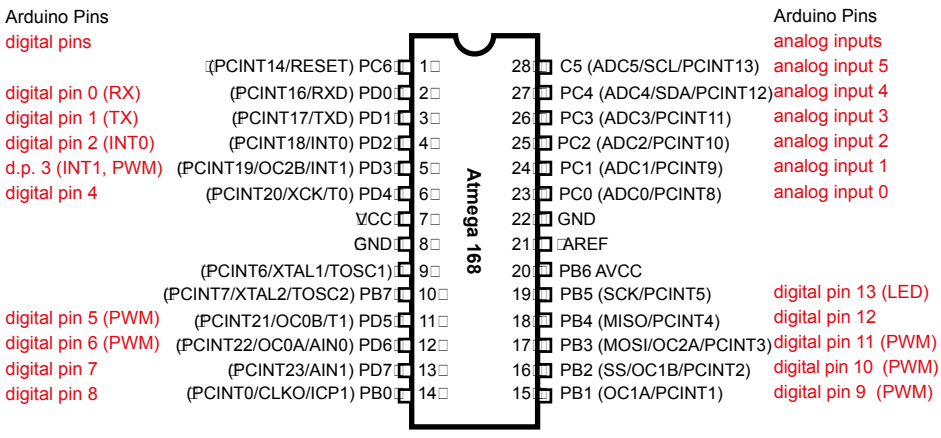
**Capacitors shown in dotted lines are not currently on the board but will be added in the next revision for Arduino 0009 compatibility.**




**Programming cable connections between a RBBB and a FTDI TTL-232R USB to TTL serial cable. (Without the capacitor hack.)**

**Please note that the TX and RX labels refer to the cable's labeling, not the chip's. The TX label at the programming connector is electrically connected to the Arduino's RX pin (digital pin 0) and the cable's RX pin is connected to the Arduino's TX pin (digital pin 1).**

-




**Pin mapping of the Atmega168 chip to the Real Bare Bones Board**

**Troubleshooting**

**Symptom: No pin 13 LED flash.**

**Causes:**

LED in backwards

Atmega168 in backwards

electrolytic capacitor in backwards

voltage regulator in backwards

no power at external input- check power adapter & polarity

bad solder connection

power supply connections reversed - check external supply with a multimeter

**Procedure:** check for 5 volts at power buses: at USB port, at power jack, near analog pins, at pins 7&8 of the Atmega168

If 5V is found at power bus pins, LED is in backwards or poor solder joint. If low or incorrect voltage, voltage regulator, solder joints, power supply

**Symptom: Power OK, but no pin 13 light flashes, and program won't download to board**

**Hardware Causes:**

Atmega168 in backwards or not seated properly (check for pins that have "escaped the socket"

Atmega168 not programmed with bootloader

resonator wrong value or missing

**Symptom: Pin 13 light flashes, but program won't download to board**

**Hardware Causes:**

bad programming cable

FTDI drivers not installed: check Arduino->Tools->Serial Port

on PC - (Comm port higher than 2 should show - choose highest comm port)

on Mac - Check for driver with "FT" in title

bad solder joint at cable connector or pins 2& 3 (check for shorts or bad (solder joints) on all pins, reheat all solder joints

wrong resonator value

**Procedure:** if you have an oscilloscope, check for signals across resonator pins and on RX pin during download.

**Software / PC side causes:**

check for FTDI drivers installed (if using USB cable)

check for proper chip (Atmega168) selected in Arduino->Tools->Microcontroller->Atmega168

click RESET switch simultaneously with download attempt

if using auto-reset hack, adjust port settings (Windows only)

Device Manager - Comm Ports - USB Serial Port - Port Settings - Advanced button - Set RTS On Close

**General "Cure-Alls" and troubleshooting:**

check orientation on all polarized parts, V.R., caps, diodes, V.R., socket and chip.

Reheat all solder pads on bottom of board, look for bridges (shorts) on chip pins

clean PCB with toothbrush and isopropyl alcohol

swap cables, power supplies if possible

Arduino is an open-source hardware and software initiative closely related to the Wiring and Processing open-source initiatives.

**Arduino Home - http://arduino.cc**

**Wiring Home - http://wiring.org.co/**

**Processing Home - http://processing.org/**

**The Really Bare-Bones Board is an open-source hardware project of Paul Badger and Modern Device Company, moderndevice.com**

Unlike other Modern Device Designs, the printed circuit board designs and schematics for the Really Bare-Bones Board are in the public domain. We hope you will only use them for good things.

**Electrical Soldering for Beginners**

**Equipment**

Use a high-quality soldering iron with the sharpest point you can find. It should be rated between 15 and 25 watts.

Keep the soldering iron tinned (coated with solder) at all times. The tip should look slivery and shiny. On a new soldering iron, it is important to tin the tip as soon as the soldering iron gets hot, the first time it is used.

Wipe your soldering iron tip off on a wet sponge, or a copper "scrubbie", to keep it clean and shiny. Do this whenever the tip stops looking shiny or has too much solder buildup on it.

Use either leaded or "no-lead" solder but be aware no-lead solder is a little harder to use for beginners, and makes solder joints that are slightly less shiny than leaded solder. That being said, no-lead solder is a good idea. Radio Shack sells small quantity packages.

Work in a room with some ventilation. There is a tiny bit of lead in solder fumes but the flux fumes are more of a health-hazard than the lead. Jameco sells a nice soldering iron / carbon filter combination for under $100.

**Procedure**

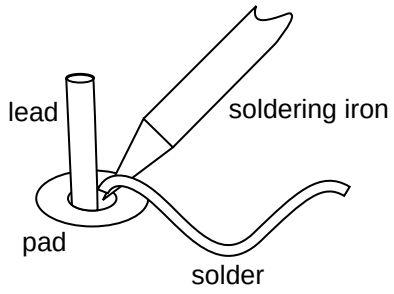
Heat the pad for about a second, then apply solder to the heated pad or leads, not the soldering iron. After the solder melts and "grabs" the pad, continue heating for another second.

If you haven't gotten the solder to grab after about 4 seconds let the joint cool down before trying again. Too much heat can ruin electronic components, but most beginners err on the side of too little heat ("cold" solder joints). If the solder joint looks lumpy, or if the solder doesn't completely cover the pad, the solder joint needs more heat. Just reheat it again until you see the flux around it "simmer" a bit, and the solder grabs the pad and smoothes out.

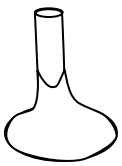
**Cleaning**

It is a good idea to clean the solder flux off your board with a toothbrush and isopropyl (not denatured) alcohol, when you are done with your board. Most fluxes when left on the board for extended periods of time, will corrode pcb pads and traces. Additionally, fluxes are not perfect insulators, so can affect the electrical operation of your circuit.

Splash a little alcohol on the board and scrub with a toothbrush. Rinse with clean alcohol and repeat until the board looks flux-free. Dry it off a little with a paper towel or rag and have a good inspection of your solder joints, to make sure pins are not bridged (shorted) with solder, and that all the solder joints look shiny, smooth, and cover the pads completely. Questionable solder joints may be fixed simply by reheating with the soldering iron.




cold solder joint
(not enough heat)



good solder joint

- smooth meniscus
- shiny
- covers pad