**Wunderlick Interactive**
Todd Bailey
Tm bailey@hotmail.com
(312) 307-9296

# Who Got a Big Ol' Butt v1.0
Synthesizer Kit

---------------
4/06
---------------
User's Manual


---------------
---------------

## 1.) Mission Statement:
----------------------

So Voltage is always rocking this aesthetic where we believe in building our own gear, most specifically electronics. It's been a personal goal of mine to advance the idea that "normal" people can build their own original instruments with a few tools and parts. I think this is especially important for noise musicians to realize, since noise-based composition is based so heavily on actual sounds, and the originality of those sounds can characterize a performer -- as opposed to a more "song" based composition process seen in rock or pop music. I believe that in the Jetsons-perfect future of music, everybody will build their own stuff.

All that having been said, going from not knowing how to solder to being able to build a synth from the ground up is a daunting task. Electronics as a field is dominated by proper engineers and the educational tools out there are typically geared towards the engineering community. On the other extreme, circuitbending DIY sites and the like abound, but it's hard to get a grasp on electronics theory and the idea that you can DESIGN your own circuit from the ground up and not just hack somebody else's. Taking all of this into account I decided to build and sell a simple synthesizer kit that I believe anybody with the drive to learn and a soldering iron can build. The tenets of the design were as follows:

1.) It should be cheap and simple, so that dirty weirdos aren't scared off by it.
2.) It should still sound cool, and hopefully unique. If you build something cheap, and it sucks, that doesn't really make a very good case for designing your own instruments.
3.) It should use components that the person building the kit can find easily, in case they fudge stuff while learning.
4.) It should be as fudge-proof as possible.
5.) Once completed, the circuit should easily allow expansion, circuitbending, and general tinkering so that the user can keep learning.

To this end, the kit uses only two ICs (integrated circuits -- chips, the black bug looking things) and both are very common and cheap. These chips are "socketed" which means they can be changed out without de-soldering them:

a.) so if you mess them up and break them you can change them easily.
b.) so that you aren't afraid to try things that might mess them up and break them.
c.) so that if you don't know if something is messed up and broken you can easily pull the chip to isolate your problem.
d.) so you can experiment by putting different chips in the sockets.

Furthermore, the "passive" components (the resistors, capacitors, potentiometers, switches and diodes) are also all common parts and with a few exceptions, are not critical values. This means that if a 10uF capacitor is called for somewhere in the circuit and you use a 2.2uF capacitor, it should still work -- and the resulting change in the sound will teach you about whatever that part actually did.

Everything is "through-hole". This means that all the parts have little legs that are easy to solder to. In the big cold world of professional nerd-dom, it isn't always nice like that.

The kit has all the parts you need to start making sound with it. It isn't all necessarily the best or most easy stuff to adjust live -- IE, tuning the kit with the trimmer pots is a little annoying, but it's a simple matter once you understand how the kit works to replace them with bigger pots with real knobs if you want to mount the whole thing in a box or chassis or the like. But they keep the kit cheap and don't require you to build anything besides just the synth if you don't want to.

The kit encourages customization in other ways, too. For instance, it is designed right out of the box to take an audio in and generate audio out, but can be easily changed to accommodate control voltages. The undedicated op-amp can be used to make an independent oscillator or filter or amplifier stage. Finally, the "heart of the circuit", the microcontroller, can be reprogrammed to make a completely different instrument. I don't expect everybody who picks up the kit to get into microcontroller programming, but it's there to do if you want to. My code, while far from perfect, is a starting place for learning how to rock digital synthesis of your own design in real time.

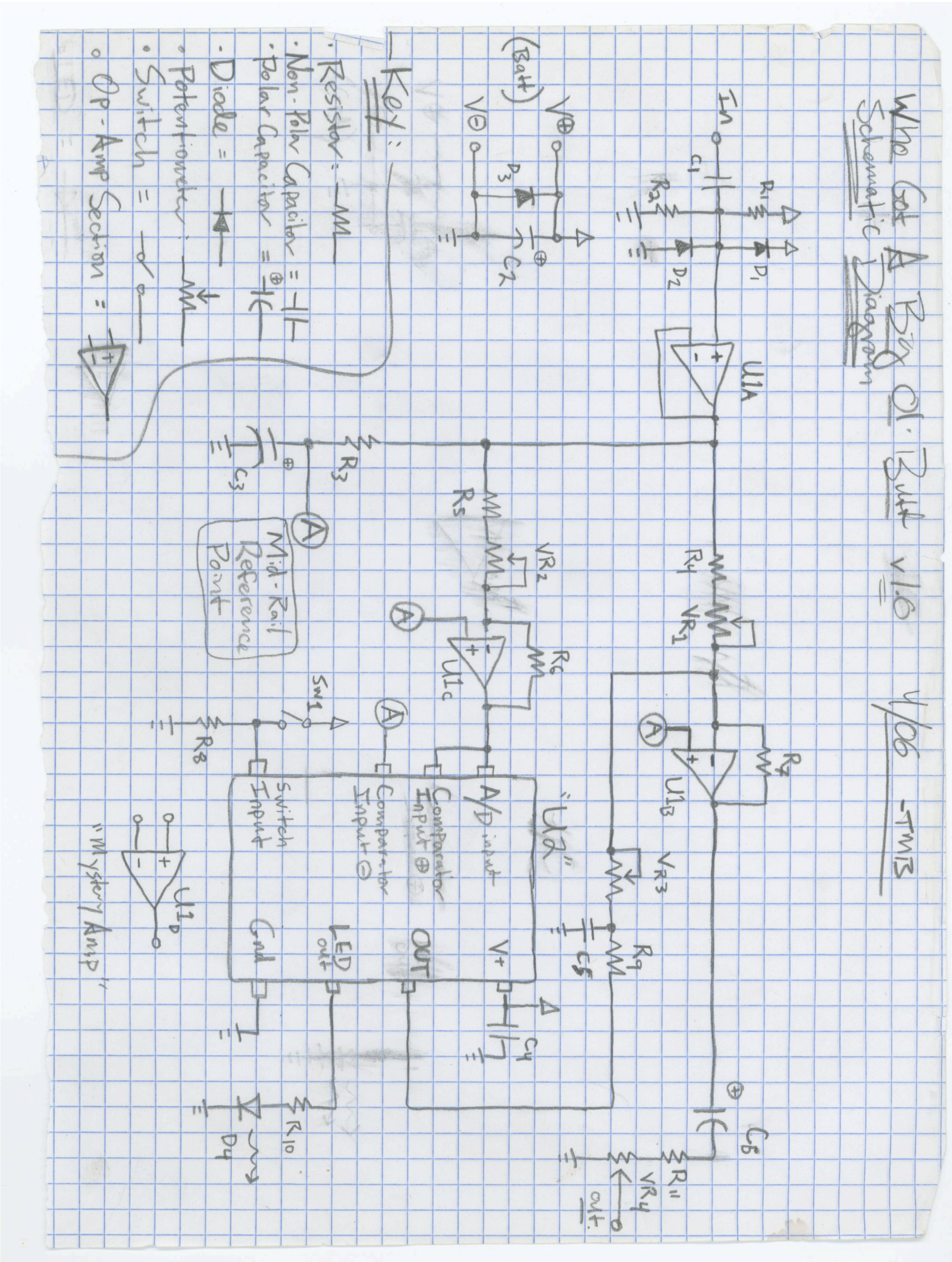
In closing, I'd also like to say that this kit is very much a labor of love on my part. As such, a lot of it got done at 4am, or in a van, or both. There are elements of it that leave something to be desired -- and if you find some, I'd love to hear about it. If there's something you wish it would do, and think that it might be able to, and want me to shoot you some code or a schematic, let me know and I'll do my best. Thanks for checking out "Who Got A Big Ol' Butt v1.0".

NOTE: If you aren't really into geeking out on electronics design and theory, and you just want to get into making noise quickly, feel free to skip the next three sections and go right to "Building the Kit". But don't come running to me if you can't get a date with the engineer of your dreams.

-----------------------
## 2.) Schematic Diagram:
-----------------------

Included below is the schematic for Who Got a Big Ol’ Butt. This shows you how all the parts are connected and in general illustrates the circuit’s functions more clearly than looking at a PCB. Very few components have a part value next to them -- they have a reference designator like “R1” or “U2”. This means that you can change the value of these parts without re-writing the schematic every time. One needs to refer to the parts list to actually assemble something from this schematic.

In general, graph paper rules the school for drawing schematics, but there are a lot of (sometimes free) schematic capture programs out there that do it as well, and produce a much more “professional” looking product that is more easily changed and can be linked to a PCB design program to minimize layout errors. I say, bump that. Pencil is great. Y’all kit fiends are lucky this drawing isn’t still on the back of the pizza box where it started.



Who Got A Big Ol' Butt v1.6 4/06 -TWB
Schematic Diagram
In
C1
R1
R2
D1
D2
U1A
V⊕
V⊖
(Batt)
D3
C2
Key:
· Resistor =
· Non-Polar Capacitor =
· Polar Capacitor =
· Diode =
· Potentiometer =
· Switch =
· Op-Amp Section =
R3
C3
A
Mid-Rail Reference Point
R4
VR1
R5
VR2
R6
U1C
R7
U1B
SW1
R8
"U2"
A/D input
Comparator Input ⊕
Comparator Input ⊖
Switch Input
V+
OUT
LED out
Gnd
VR3
R9
C5
C4
U1D
"Mystery Amp"
C6
R10
D4
R11
VR4
Out

The component values are as follows:

| Part Reference Designator: | | Part Number: | What is this really? |
| --- | --- | --- | --- |
| U1 | | MCP604 | Quad Operational Amplifier |
| U2 | | PIC12F683 | 8-bit microcontroller |
| D1-D3 | (polar!) | 1N4004 | 1A 400v Rectifier Diode |
| D4 | (polar!) | ?? | Some LED or other. |
| Sw1 | | SW403-ND | Momentary NO Tact Switch (Digikey) |
| VR1,VR2 | | 3362H-1-103 | 10k Trimmer Potentiometer |
| VR3 | | 3362H-1-105 | 1M Trimmer Potentiometer |
| VR4 | | 3362H-1-102 | 1K Trimmer Potentiometer |
| C1 | | BC1621-ND | .1uF Film Capacitor |
| C2,C3,C6 | (polar!) | 493-1279 | 10uF Electrolytic Capacitor |
| C4,C5 | | 399-1943 | 2200pF C0G Ceramic Capacitor |
| R1,R2 | | 470k | 470k 1/8W 5% Resistor |
| | | (Color code: | Yellow, Purple, Yellow, Gold) |
| R3, R6-R8 | | 10k | 10k 1/8W 5% Resistor |
| | | (Color code: | Brown, Black, Orange, Gold) |
| R4,R5 | | 470 Ohm | 470ohm 1/8W 5% Resistor |
| | | (Color code: | Yellow, Purple, Brown, Gold) |
| R9,R11 | | 1.2K | 1.2K 1/8W 5% Resistor |
| | | (Color code: | Brown, Red, Red, Gold) |
| R10 | | 100 Ohm | 100 Ohm 1/8W 5% Resistor |
| | | (Color code: | Brown, Black, Brown, Gold) |

All part numbers for the switches, caps and pots are Digi-Key (a electronics component mail order house, www.digikey.com) stock numbers. The other parts are generic descriptions or the name of the part. They're all easy to find online, and most can be found at Radio Shack.

## 3.) Theory of Operation:

AN IMPORTANT NOTE TO THE KIT BUILDER:

Please note: I've tried to make as much of this as I can in both "dude talk" and "engineer-speak". The former is so that the layperson who doesn't understand electronics at all can actually get an idea of what is going on with the circuit, and the latter is because a.) ultimately, once you're a G, engineer-speak is more precise and clear, and b.) so that the layperson can get some familiarity with technical writing -- if you want to get into building your own stuff, it's both hard to avoid and also fun. For the true electronics heads out there, forgive me. In the dude sections I tend to generalize a lot and sometimes gloss over the absolute truth for an approach that I feel is easier to understand. Also, if you've never done a single thing with electronics before, don't be discouraged if you don't understand all of this the first time you read it. If you really want to, you'll get it, and it may take time. If you really want to understand (and not just build) this circuit, read this section and study the schematic, and do it many times. Build the circuit and conduct some experiments changing values and removing components and the like. Use Wikipedia to look up things you might not understand, like "resistor". Find reference books and web pages.
That's how I learned!

If you're really curious I can't stress enough how important it is to get some books. If you're lost by all of this, reading somebody else's work can help a lot. Plus, this manual is light on photos and isn't the best soldering guide. The Ghazala book covers that really well, as do a few places on the internet. And once you've got the simple stuff licked, things get really exciting with Horowitz and Hill and the rest of them Franchize Gentlemen.

On with the show:

"Who Got a Big Ol' Butt v1.0" is a synthesis kit which begins by taking an analog signal as an input. The signal is buffered by an operational amplifier and then digitized via the Analog-to-Digital converter of a microcontroller. Once in the digital domain, the microcontroller analyzes the signal mathematically, and generates an audio output based on a program in the synth's firmware. The synth also passes along a copy of the unaltered signal to the output, so that the two signals can be mixed together. The programs in the were written by me and are not code protected -- if the user has the tools and inclination they are free to "dump the code" and change it to their hearts' content.

All this having been said, let's start analyzing the circuit, starting with the input.

--Engineer:

The signal enters the circuit at the left side of capacitor C1. Capacitor C1 and resistors R1 and R2 form a bias network, which allow the input signal to stay centered in the middle of the power supply rail. With the supplied component values (470k resistors and a .1uf Cap) the lowpass filter point of the input circuit is approximately 3.5 Hz -- well below the audio spectrum but high enough to produce some amount of phase-shift in very low bass frequencies. The input impedance of the circuit at normal signal amplitudes (i.e., diodes not clipping) is roughly 235k, which is a high enough impedance to allow an input from passive signal sources (like a guitar). Diodes D1 and D2 provide input protection, by clamping the voltage at the operational amplifier input to (supply voltage-0.6v) < (signal) < (supply voltage + 0.6v).

--Dude:

We only have one set of batteries to work with (there are only two wires coming from the battery pack, red and black). If you're using the battery pack that came with the kit to power the kit, it will be supplying roughly 4.5 volts. By convention, the black wire is "ground", or zero volts, and the red wire is 4.5v. For the chips to be able to "look at" a signal without freaking out, the signal must be between 0 and 4.5 volts. For most things you'd plug into the kit, that isn't a problem. Even very hot output signals from other (commercial, mind you -- circuitbent electronics got that crazy style) electronics are usually less than a volt. The diodes are in this circuit to make sure that if your stoner ass plugs your speaker output into the kit or something equally muddleheaded, the excess voltage is shorted to the supplies and doesn't blow up the op-amp. In this example, it would eventually probably blow up your bass amp or the diodes, or both. And maybe then the op-amp, too. But it'd probably sound pretty killer for a second, dude!

The resistors and cap are there to address this "voltage problem" also. The input jack has two wires coming off of it, right? One of them (lets call it black, although it could be any color, and it's usually by convention the contact that comes from SLEEVE of the quarter inch plug, and not the tip, but again, either way will work) is a reference, or GROUND connection. This means that if you were measuring the signal coming into the jack, you'd measure it in relationship to GROUND, or that wire. In other words, the input signal is measured BETWEEN those two wires. If we clip that "black" wire to the circuit ground, which we've established is zero volts, then the input signal swings around that point, or zero volts. What's the big deal, you say? Well, if the input voltage is one volt, and it's swinging up and down (musical signals always swing up and down -- they are "AC signals" or Alternating Current, which means that they swing around over time. A "DC" (Direct Current) voltage doesn't swing with time. The batteries are a DC voltage source) AROUND zero volts, then sometimes as it swings, the input signal will be above zero volts, and sometimes the signal will be below zero volts. Remember then, that for the chips to work they must see a value between zero and 4.5 volts, so when the signal is swinging below zero, the circuit won't work.

Enter the resistors and cap. The cap blocks DC current from the outside world. This makes the signal after the cap "float", meaning that it still swings like an AC signal (like music) but it NO LONGER SWINGS AROUND ZERO VOLTS. As a matter of fact, if only the cap were in the circuit, and not the resistors, the signal would not swing around any defined reference voltage at all, and would literally float off into space. Or at least mess things up good in the circuit. The resistors re-establish a relationship to a DC voltage for the signal. It's enough to explain this by saying that the two resistors are equal in value and one is tied to the positive supply (4.5v) and one is tied to zero. Therefore, the "DC reference point" here is half-way between 4.5 and 0, or in this example, 2.25v. This means that even swinging a total of 2v, a signal only ever gets as positive as 3.25v and as negative as 1.25 (i.e., 3.25-1.25 = 2) which is still safely within the range of the input of the op amp.

Note, one way to think of DC is as "the very lowest frequency there is". By making a network which blocks this very low frequency (the RC network we just talked about) we inadvertently risk blocking a little bit of bass in the audio region as well. Which is the greatest of all sins. I've calculated the values of this bias network to try and prevent this -- but changing these values will change the value of this "high pass filter", and it will change the "input impedance" too. See if you can figure out how...

On to U1A, the first Op Amp Section:

--Engineer:
Operational Amplifier U1A is configured as a unity gain buffer, isolating the input and providing a low impedance drive to the following sections.

--Dude: U1A is really simple, provided you know what an operational amplifier does. In short, an op-amp is a simple, powerful, easy to use amplifier building block. Inside the MCP604 there are actually four of these little gems, all running independently. Op amps can be used to make, duh, amplifiers -- but also filters and oscillators and analog math circuits. The op amp follows a few easy rules, is cheap, and can be very forgiving. If you want to get into building analog circuits, a great place to start is the op-amp. See the Mimms book if you're curious about some easy op-amp projects.

Suffice it to say that U1A in this circuit is acting as a "buffer", which means an amplifier that doesn't add any voltage gain. What, no gain? That's no amp! Well, actually, what it provides is "power gain", meaning that although the signal isn't any larger, it is in effect "stronger". If you just plugged your guitar straight into the mess of resistors after U1a, it would leech the signal a lot, and where the knobs were would affect the shape of your guitar signal, etc etc. With the op amp in place, the guitar sees something "easy to push" and the op amp "pushes" the circuit components after it a lot more "firmly".

All this is hinting at an important parameter of electronics: current (and its friend, impedance). Understand current, and you've got electronics licked.

On to R3 and C3:

--Engineer: R3 and C3 form a low-pass filter network which strips the audio component from the DC level after U1a (at point A) to provide a low-impedance mid-rail reference point for the microcontroller and amplifier circuits. Including the filter after U1A allows the mid rail point to account for any offset voltage in U1a, and allows only one voltage divider (R1/R2) to provide the mid rail point for the entire system, to eliminate differences that could be introduced by component variation if one used separate dividers.

--Dude: So we did all that work before to get the audio input riding on 2.25 volts, and now we're taking that audio back off the DC voltage. Why? Because we need a reference for the all the other sub-circuits in the synth, for the same reason we needed one for the first op-amp. We've already gotten a "mid rail reference point" (i.e., a DC voltage half-way between the positive battery voltage and ground) once -- in fact it was the very first thing we did in the circuit with R1 and R2. That 2.25v DC is still there after U1a, but so is the audio signal. R3 and C3 form a "low pass filter" which eliminates the audio signal at point A and leaves the DC to be used as a reference by the other circuits.

There are other, and probably better ways, that I could have designed this reference, but I thought it was cute doing it this way. Time will tell. If you can think of a better way (which you'll be able to without much trouble if you really nerd out) I'd love to hear it.

R5,VR2,R6 and U1c:

--Engineer: Operational Amplifier U1c is configured as an variable-gain inverting amplifier. The gain varies from .96 to 21.3 as VR1 is swept from end to end. The output of U1c drives the Analog-to-Digital Converter and Comparator Inputs on the microcontroller.

--Dude: Here's where the magic starts to happen. The audio signal from after U1a is passed to this amplifier circuit, which feeds the microcontroller. A/D converters are finicky about the kind of signal they want, so this amplifier does two things. First, it feeds the A/D the low-impedance input that it wants and second, it provides (voltage) gain. Meaning that as you turn up the knob (or potentiometer, more exactly) VR2, the signal gets bigger. The amount of signal going into the A/D determines the "sensitivity" of the circuit. Meaning, as you turn up the level into the microcontroller, the synth is more likely to do whatever it is that the particular program is telling it to do. The "sensitivity" pot (VR2) is labeled "Tresvant" on the printed circuit board, after Ralph.

U2

--Engineer:
I love that band!
--Dude:
Didn’t Paperrad make a video about that Ipod?

--Engineer: U2 is a PIC12f683 microcontroller running audio synthesis firmware. The buffered and amplified audio in is fed to the A/D converter and comparator inputs, and processed by the onboard program. Switch SW1 provides user input to change and toggle programs. C4 provides local supply bypassing. D4, the LED, is an indicator of the status of the circuit. The output of the digitally generated signals is passed to the final summing amplifier stage at U1b.

--Dude: Don't break this chip, because as per the mission statement, they're cheap and easy to find HOWEVER this one has a program that I wrote on it. Meaning if the cat eats it, either I have to email you the code and you have to burn a new chip (which takes some special equipment) OR I have to mail a new one to you.

All that having been said, the PIC is the magic lamp wherein the genie resides in this particular circuit. The beautiful thing about a digital synth (versus analog) is that to change the sounds the synth makes, you don't necessarily need to change the CIRCUIT, you can just change the program. The soldering iron stays off. And since this chip has enough memory to store multiple programs, the synth can be changed into something completely different without repatching or touching any knobs at all. I'm not arguing that digital synthesis is better in any way than analog synthesis (in many ways it isn't) but I am saying that it allows more flexibility with the same hardware. The PIC on this board is in the neighborhood of $2, as is the op-amp, and for that few parts for that little dough, you can get a device which does a lot MORE stuff by programming than you could with all analog components. In general, I think the whole analog / digital debate, at least w/r/t to audio, is kind of silly -- both are wonderful mythologies with great stories.

So what does it do?

(If you're feeling brave you may wish to consult the PIC12f683 datasheet. The link is included here for reference. There's a lot going on in there, so don't feel bad if you don't follow it. But there's a wealth of information there)

If you look at the schematic, you'll see that the PIC takes two inputs from U1c -- the "A/D input" and the "Comparator Input +". Remember that these are copies of the audio in, after amplification. There is also an input from the mid-rail reference point at "Comparator Input -". Let's start by talking about the Comparator.

Inside the PIC is a device which can look at two inputs and tell you which one is "bigger", or more precisely, which is the higher voltage. The device "compares" the two inputs to generate an output. We can see both inputs to the comparator on the schematic -- Comparator + and -. Now, we know that the mid rail reference point is just a DC voltage halfway between the positive battery terminal and the negative, and we also know that it is the center point around which the audio signal floats. And we know it's connected to "Comparator -". The other input, Comparator +, is connected to a copy of the audio signal after it's been amplified by U1c. This means that the PIC can tell which voltage is higher -- the center-rail reference point or the signal. Essentially, this tells us whether the waveform is "above or below" the reference, or more importantly, when and how many times the audio input "crosses zero". These zero crossings tell us the frequency of the input signal. This is important for doing any kind of frequency-related synthesis, for instance, generating an OCTAVE DOWN signal by simply dividing these crossings by two. Follow?

The next input, the A/D input, allows a simple approximation of what Pro-Tools or any other advanced digital signal processor does. "A/D" stands for "Analog to Digital Converter" and it takes your audio signal IN and generates a NUMBER that the microprocessor can then analyze mathematically. The microprocessor shuffles these numbers however the program tells it to, and then spits these back out into the real world as a waveform of its own. The A/D converter periodically takes a snapshot of the voltage at it's input. How fast it takes these snapshots is called the "sample rate". The number generated is related to what the voltage at the A/D input is at the point that the sample is taken. The A/D converter has reference points as well, like the comparator, but it uses ground and the positive battery voltage as the references and doesn't care about the midrail reference point (well -- it almost doesn't care. The audio signal into the A/D input is still centered on that midpoint -- but that midpoint doesn't have any special meaning unless the software dictates that it should). Meaning, that if the A/D converter can generate numbers from 0-100, and the input voltage is at 0v (the same as ground, or it's low reference point) the number it would generate would be 0. If the input voltage was equal to the positive battery voltage, the A/D would generate the number 100. In between, the A/D converter's output follows the voltage on the input in steps equal to (positive reference voltage - negative reference voltage (ground) / possible numbers the converter can generate). Or in this example (4.5 - 0 / 100). So every .045v of change on the input moves the A/D's output number up or down. The number steps the A/D's output has is called its "resolution", because it reflects how minutely the A/D can keep track of an analog signal.

If an A/D only had a resolution of "2" it would be a comparator! Think about it. For reference, an A/D's resolution is usually referred to in "bits" -- our A/D converter is an 10-bit converter, which means that it can differentiate 1024 point in between the positive and negative rail -- so the A/D can see a voltage change as small as .004 volts. I throw out the last two bits in software because I'm nasty like that, and it makes the math easier. This gives us an effective resolution of 8 bits.

In general, when thinking about A/D conversion, the "sample rate" is how FAST the converter can track a signal and the "resolution" is how well. To do a good job in audio you need both. To do something artsy-fartsy, you can slack pretty hard. Another general rule of thumb is that the greater the resolution and sample rate are, the more expensive the part is AND the more data the converter spits out, which means that dealing with it takes a.) a good programmer AND/OR b.) a very fast processor, or most likely both.

The output of U2 is a digital output. That means it can only do three things -- turn off, send a "high" voltage out, equal to the positive supply rail, or send a "low" voltage out, equal to the negative supply rail. This means it can only generate square waves. You can do a lot with square waves, as you'll see if you keep messing with the synth. Not only can you make that characteristic buzzy sound, you can make random noise, and you can even play bad audio samples!

The LED and switch are self-explanatory. The switch is your way of talking to the PIC, and the LED is its way of telling you what's on its mind. The LED is a good example of how you can use a square wave to simulate something other than "on/off". If you look at the LED you can see it fade up and down. How can it do that if a digital output can only go "high" and "low"? Wouldn't it just be all the way up or off? How could it fade? The answer: The lamp IS either fully on or off at any given time -- it just alternates faster than your eye can see. The amount (ratio, actually) of time it spends "high" versus "low" gives the illusion of fading. Neat, huh? This technique is called PWM (pulse width modulation) and gets used in digital electronics all the time.

The capacitor hanging off the V+ input is just to keep the power into the PIC clean. This is called a "bypass capacitor". They aren't very glamorous, but they're really important to keep your circuit from doing strange things. That you didn't want it to, I mean.

R9 and C5:

Engineer:
R9 and C5 form a low-pass filter which eliminates digital artifacts above the audio range on the output.

Dude:
This sub-circuit is just like R3 and C3 earlier, except it doesn't cut AUDIO, it cuts all the stuff that the PIC kicks out above the audio range. Is this really important? Not really, it's just a "cleaner way of getting noise". Feel free to remove or change the value of C5, and the tone of the synth will change a lot. Keep R9 in there or funny stuff will happen when you turn VR3 all the way up.

VR3, R4, VR1, R7 and U1B:

Engineer:
Op amp U1b is configured as an inverting amplifier which sums the outputs of the audio input buffer and the microcontroller. VR1 and VR3 allow separate level control of the unaltered audio in and synthesized audio output, respectively.

Dude:
We've already seen an inverting amplifier once like this in this circuit at U1c. This one is special because it is a "summing amplifier" which means it takes two separate audio signals adds them together. In this case, these signals are the audio input, as taken from U1a and the output of the PIC. Just like before, the value of the resistance of VR1 (Yngwie) and VR3 (Glass) and it's relationship to the fixed resistor R7 set the gain of this amplifier subcircuit. If you look up "summing amplifier" or "inverting amplifier" in either a reference text (like Horowitz and Hill) or the internet, there will be dozens of explanations of how this works exactly.

C6, R11, VR4:

Engineer:
The RC network formed by C6, R11 and VR4 forms a high-pass filter which eliminates the DC component from the audio signal and also provides "master volume" control over the synth. Since VR4 dominates the output impedance of the synth, the value of this potentiometer must be kept relatively low, necessitating a fairly large filter cap at C6 to prevent phase-shift in the low audio range.

Dude:

These are the components that handle getting your summed synthesized / input signals back out of the synth. Remember that all the signals after the first resistor network R1 and R2 are floating on a DC voltage -- that old mid-rail reference point. That's useful to our synth, but bad in the rest of the world. The capacitor C6 blocks DC to the output, and inadvertently also filters out some low frequency as well. Since loss of bass is the greatest heresy that an engineer can commit, the values of this network must be calculated carefully. The value of VR4 is kept small (1k) because it is too big, the device will be prone to noise pickup and not really be able to drive other circuits very hard. R11 is there for two reasons: a.) so that when you accidentally short the output, the op amp isn't trying to deliver infinite current into no load (and breaks) and because the output range of the circuits in this synth is 0-4.5 volts, whereas the "line level" for most other circuits is .7v. This means that if this resistor wasn't there, useful range of adjustment on VR4 would be very small, since the great majority of the throw of the pot would be "loud". And sometimes, somebody might want to be quiet. Remove or reduce R11 to make your synth louder, but don't short it with the volume all the way up.

U1D:

Engineer:
Op amp section U1d is unused.

Dude:
WHOA! A whole extra op-amp with the pins brought out to pads on the board! This would be really great for building an analog oscillator or an extra amplifier or distortion and it seems like it's there just inviting circuitbending...

---------------------------------------------------------------------
## 3a.) The Absolutely Simplest Way I Can Explain How This Synth Works:
---------------------------------------------------------------------

Who Got A Big Ol' Butt takes an audio input and sends it to a simple, tiny, hard-working computer. This little computer looks at the audio in and generates its OWN output audio signal out based on a set of parameters in its program. The user can control how much signal the computer sees, and how much signal from any of these different sources gets to the output of the synth with the different level controls. The user can scroll through the different programs by holding the tact switch down, and can turn the synthesized output on or off by tapping the same switch.

----------------
## 4.) Source Code
----------------

Included with this kit is the assembly language source for the microcontroller. The file will be lurking around somewhere wherever you found this manual and will have the extension .asm. It is too big of an undertaking to try and explain it all, but most of it has been "commented" extensively, which means it's full of little notes I wrote to myself and to other programmers. If you REALLY want to change your synth, you'll have to get jiggy with this. Get a copy of MPLAB from Microchip (http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002)and the datasheet for the PIC12F683 and the Microchip Mid-Range Reference Manual. There's definitely a learning curve if you haven't done this sort of thing before, but it's really rewarding.

----------------------------------------------

## 5.) Assembly Instructions: "Building the Kit"

----------------------------------------------

This is the part that none of you will skip, so I'll try and do a good job. SKIM THIS ALL THE WAY THROUGH ONCE BEFORE YOU START TO WORK.
First things first --

What do I need to build this kit?

1.) A soldering iron.
2.) Solder.
3.) Diagonal Cutters (wire cutters, nippy cutters, etc -- something that'll cut wires and component leads).
4.) A small screwdriver -- phillips or flathead for adjusting the potentiometers.
5.) Three AAA batteries.

What else would help?

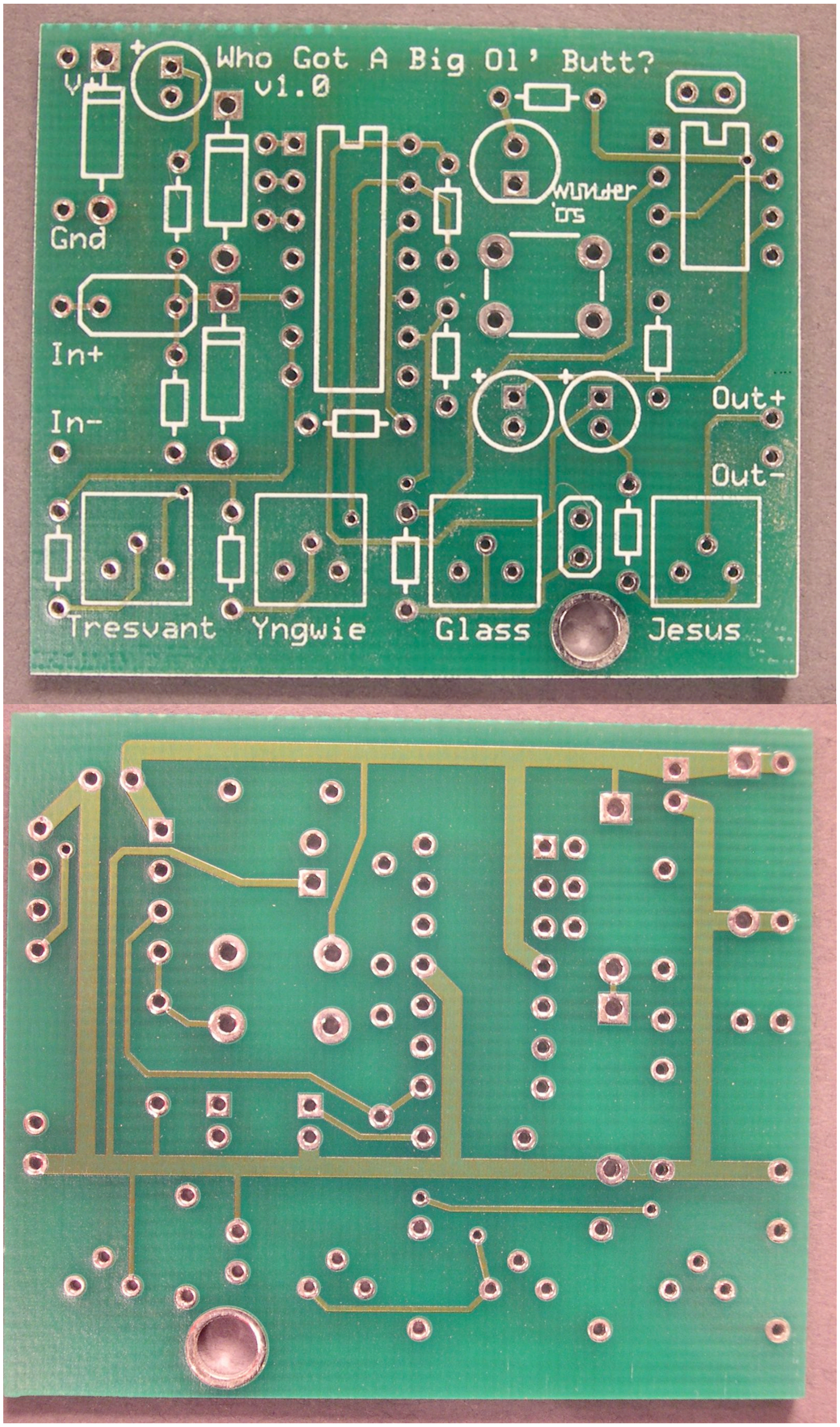
1.) Wire Strippers. These help a whole lot.
2.) A wet sponge to clean the tip of your iron. I can't tell you how much easier this will make your job, and it's so cheap.
3.) Desoldering braid for when you mess up.
4.) Some rubbing alcohol and a toothbrush to "deflux" your kit when you're done.
5.) Needle nose pliers.
6.) Burn ointment.
7.) Patience.

All of the theory stuff would make a lot more sense if you can get your mitts on a digital multimeter and an oscilloscope, but you don't need them. Below is a picture of some of the nerd-ware on my workbench:


Weller
WTCPT
Output For TC201T
Series Irons Only
Use Only Original
Weller Tips.
Xcelite U.S.A.
170M
Primer

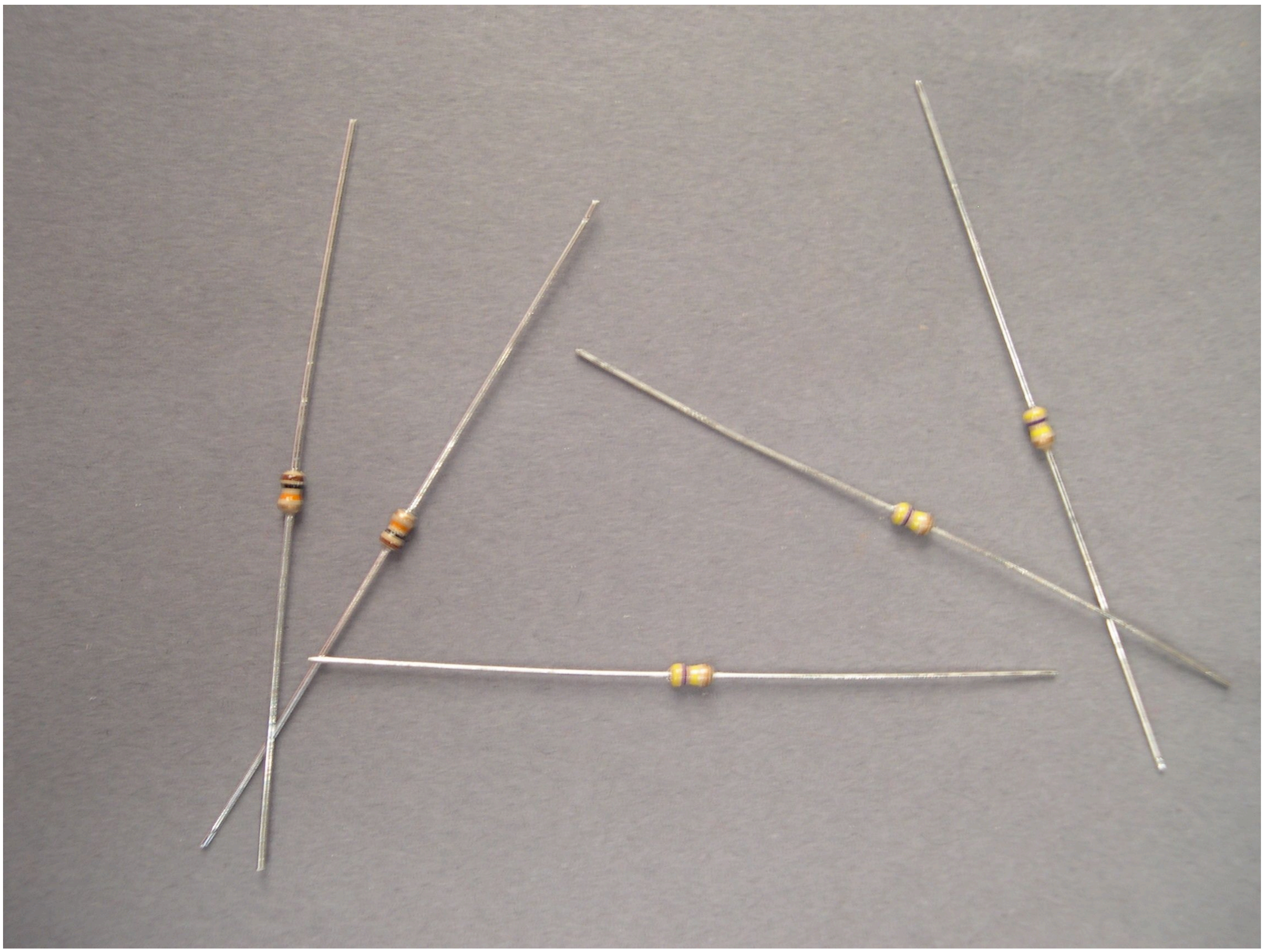
Now, what are all these components?
------------------------

Printed Circuit Board:
-----------------------



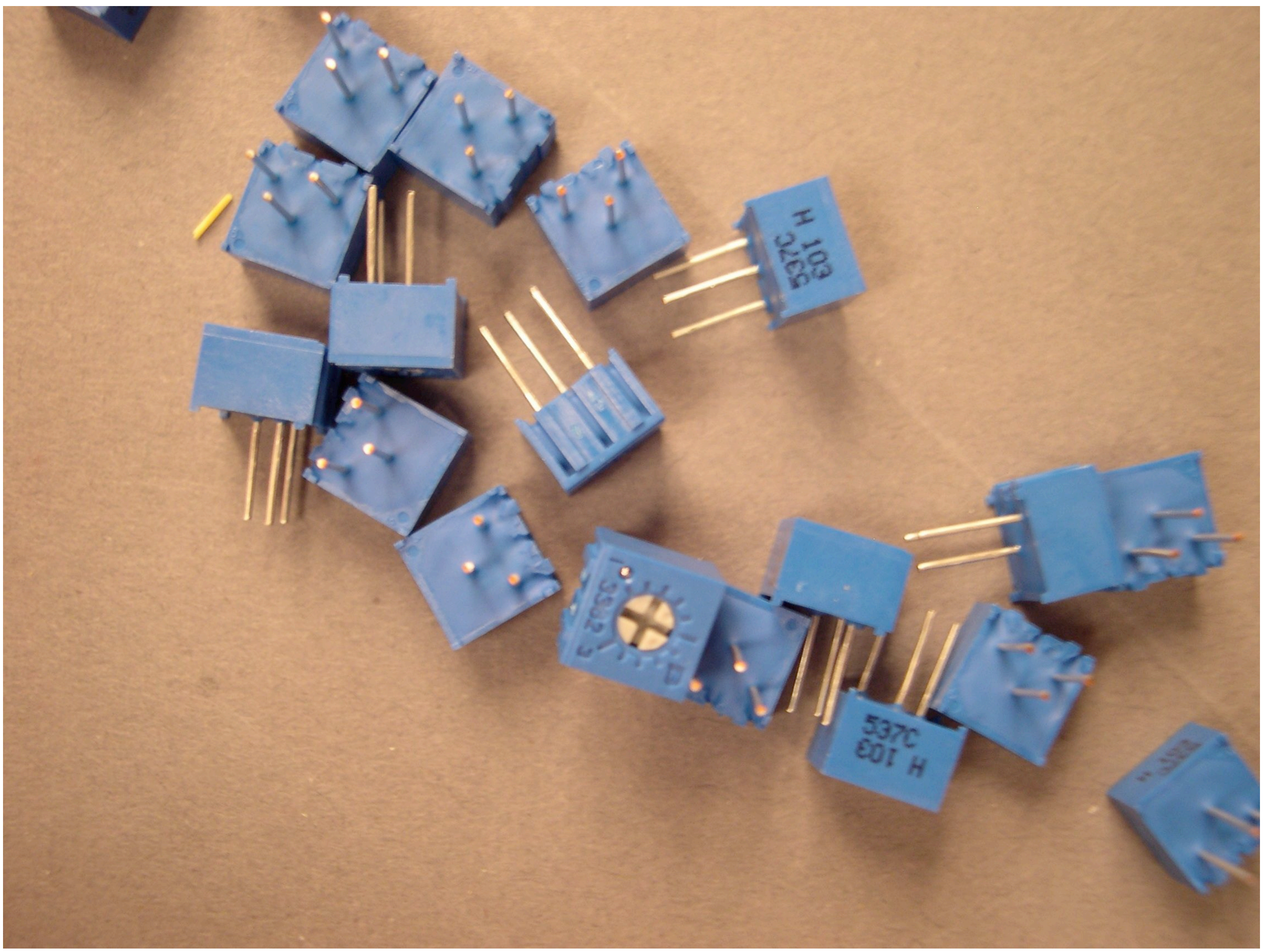
The PCB is a canvas on which you lay out your art. Anybody can make one. Being particularly punk rock involves using a bunch of tape and acid and copper, but the one in the kit was something I laid out using a free CAD program, available from www.expresspcb.com. The traces and pads on the board are made out of copper, which is adhered to an epoxy-glass laminate (a piece of fancy plastic, essentially). The copper is covered in tin (solder, essentially) that keeps it from oxidizing. The green stuff (the solder mask) is a coating that protects the copper and keeps solder only on the areas you want to solder to, and the white text is a silkscreen that helps the user align the components correctly and contains corny jokes. The big hole is for a 4-40 mounting screw.

Resistors:
-----------



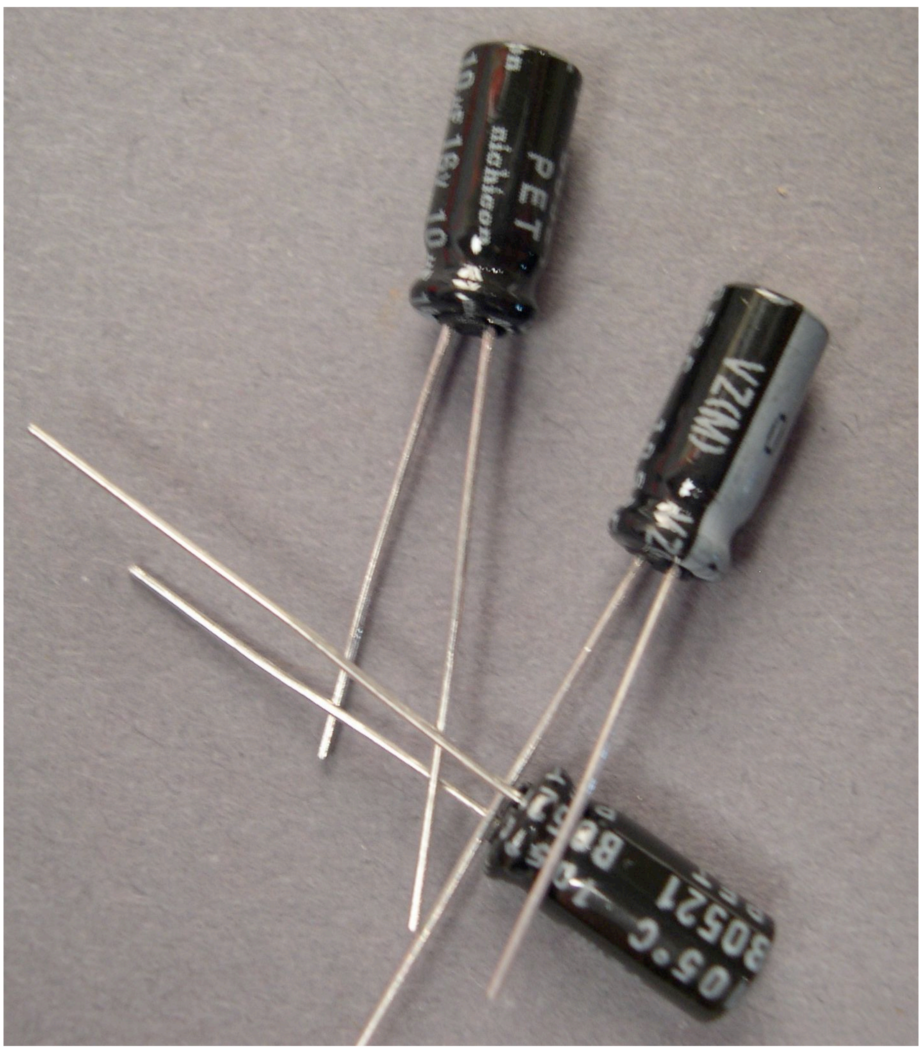
Resistors are the little workhorses of electronics. They usually consist of carbon or metal film, and they essentially oppose the flow of current. Resistors are used to divide voltage, limit current, make noise sources, and are in every electronic circuit there is.  They are really cheap.  Resistor values are measured in Ohms, the unit of resistance.  You can determine the value of the resistor by either measuring it with a multimeter OR you can read the colored bands on the body, which indicate the value of the component using a color code.  I've included the codes in the parts list for the resistors in the kit, and included the appendix on the resistor color code from Horowitz and Hill as well.  Shh, don't tell.  If you are still unclear you can google "resistor color code", and Radio Shack sells (sold?) a cheat sheet or little wheel you could buy.  An example: the resistors on the left in that picture are 10 Kilo Ohm resistors, or 10k.  The brown and black bands together mean 10 (brown being one and black being zero), the orange band means "x1000" (orange means three, and in the third position that takes the first number (ten) to the third power. The gold band indicates a 5% tolerance, meaning the actual resistor if measured should be within 5% (within 500 ohms) of 10k.

Trimmer Pots:
--------------



A “potentiometer” is just a resistor whose value you can change by turning a knob. In addition, potentiometers (or “pots” as we say in the business) have a third terminal (recall resistors have two) connected to an internal part called a “wiper”. Two of the terminals on a pot are across the internal resistor – say 10k like in this photo – and stay that way no matter how the knob changes. The wiper is connected to the knob, and as you turn the knob, the wiper slides over the resistive element and gives you a resistance value that corresponds to the wiper’s position. For example, with the knob in the dead center of a 10k pot, the resistance from outside terminal to outside terminal is 10k, and the resistance from either side of the pot to the wiper is 5k – right in the middle. “Trimmer” just means these pots are small and you can’t adjust them with your fingers. The 1k pot has a “102” on the side, the two 10k pots have a “103” on the side, and the 1M pot has a “105” on the side.

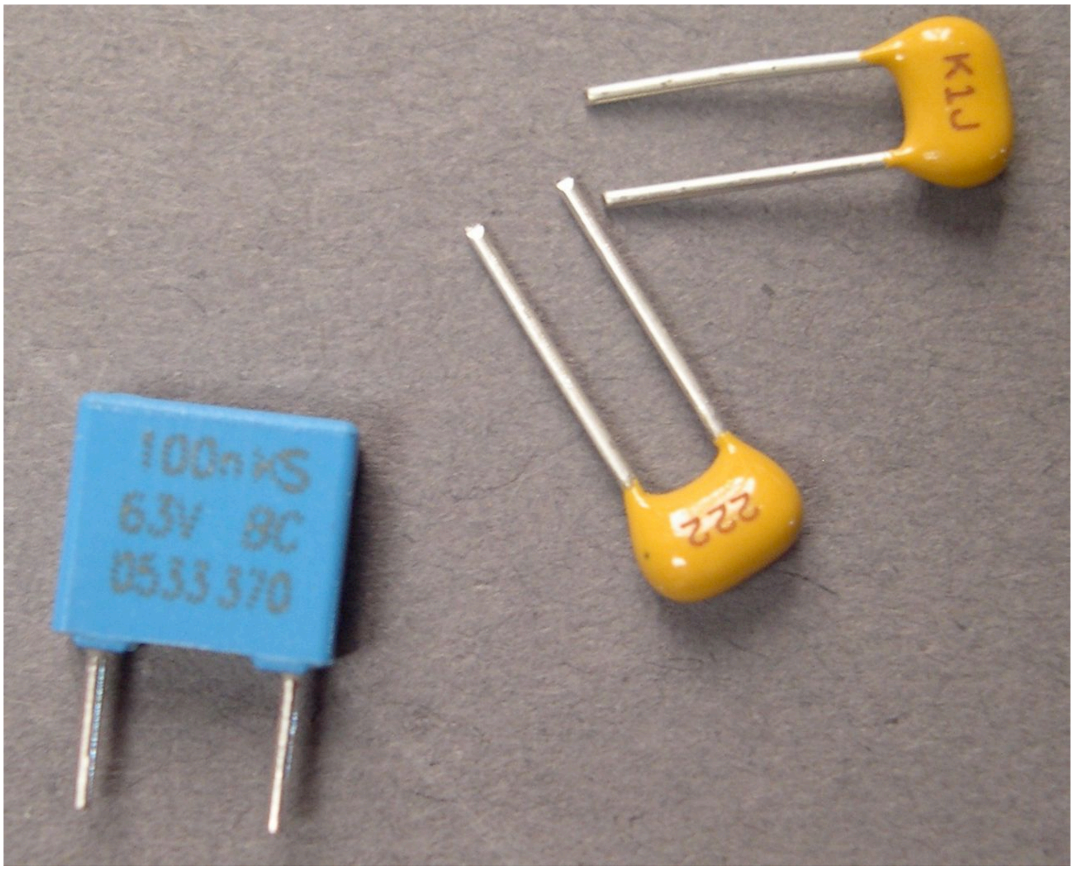
Capacitors, Polar:
-------------------




Capacitors are complicated little passive devices that introduce the element of time into a circuit. You can think of them as “charge reservoirs” or “frequency dependent resistors” but really they are their own little world unto itself.

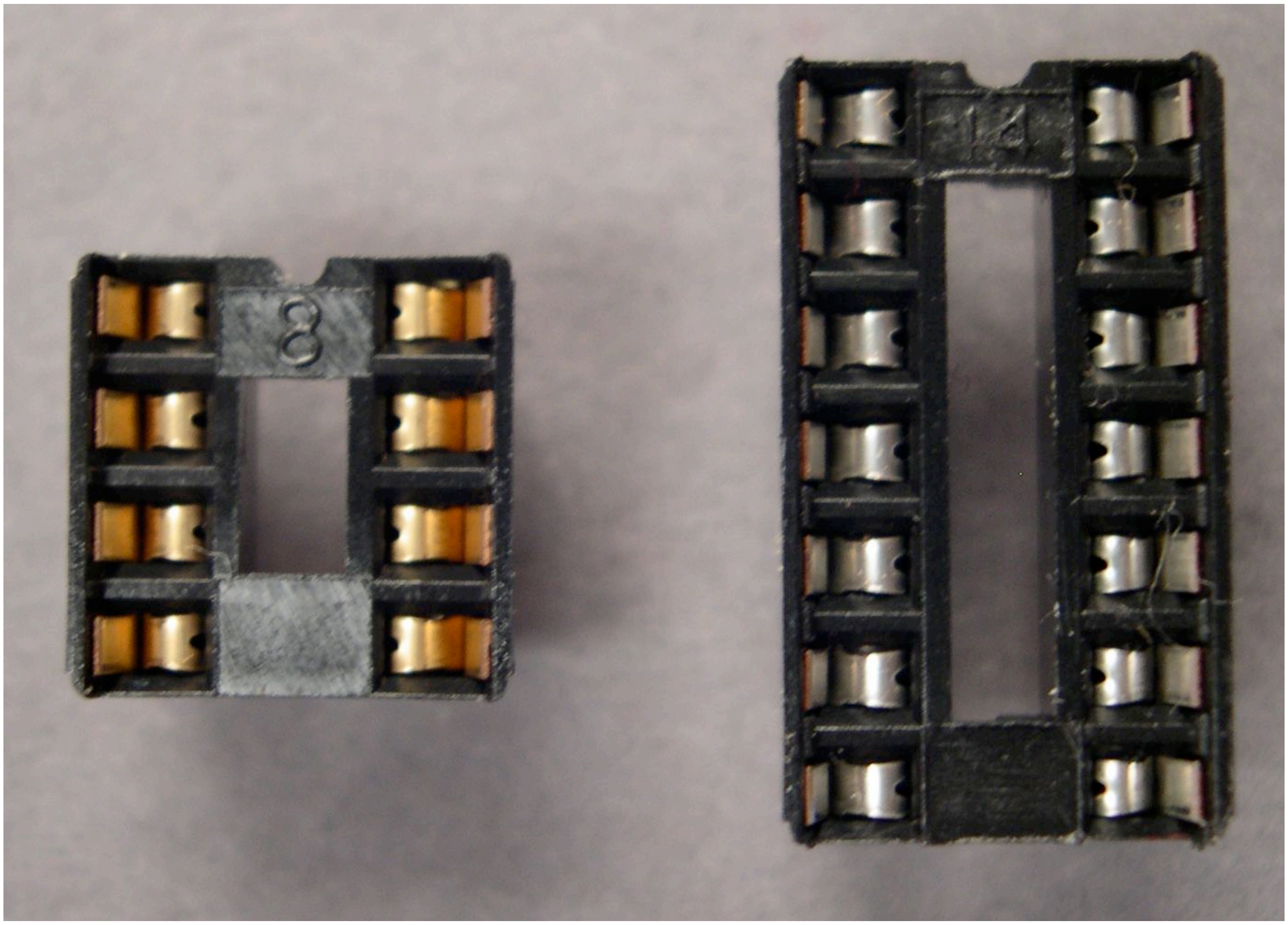
Analog nerds always love caps. These caps are called “electrolytic caps” which refers to the chemistry in their guts. Among other things, this chemistry makes these capacitors POLAR, meaning they only work when the average voltage across them is biased in one particular way – meaning that one side is always at a more positive potential than the other. What this means for us is that THEY SHOULD ONLY GO IN THE BOARD ONE WAY, even though they fit in both ways. The gray stripe is above the more negative terminal, and should go into the board AWAY from the little plus (+) sign.

Capacitors, Non-Polar:
-----------------------



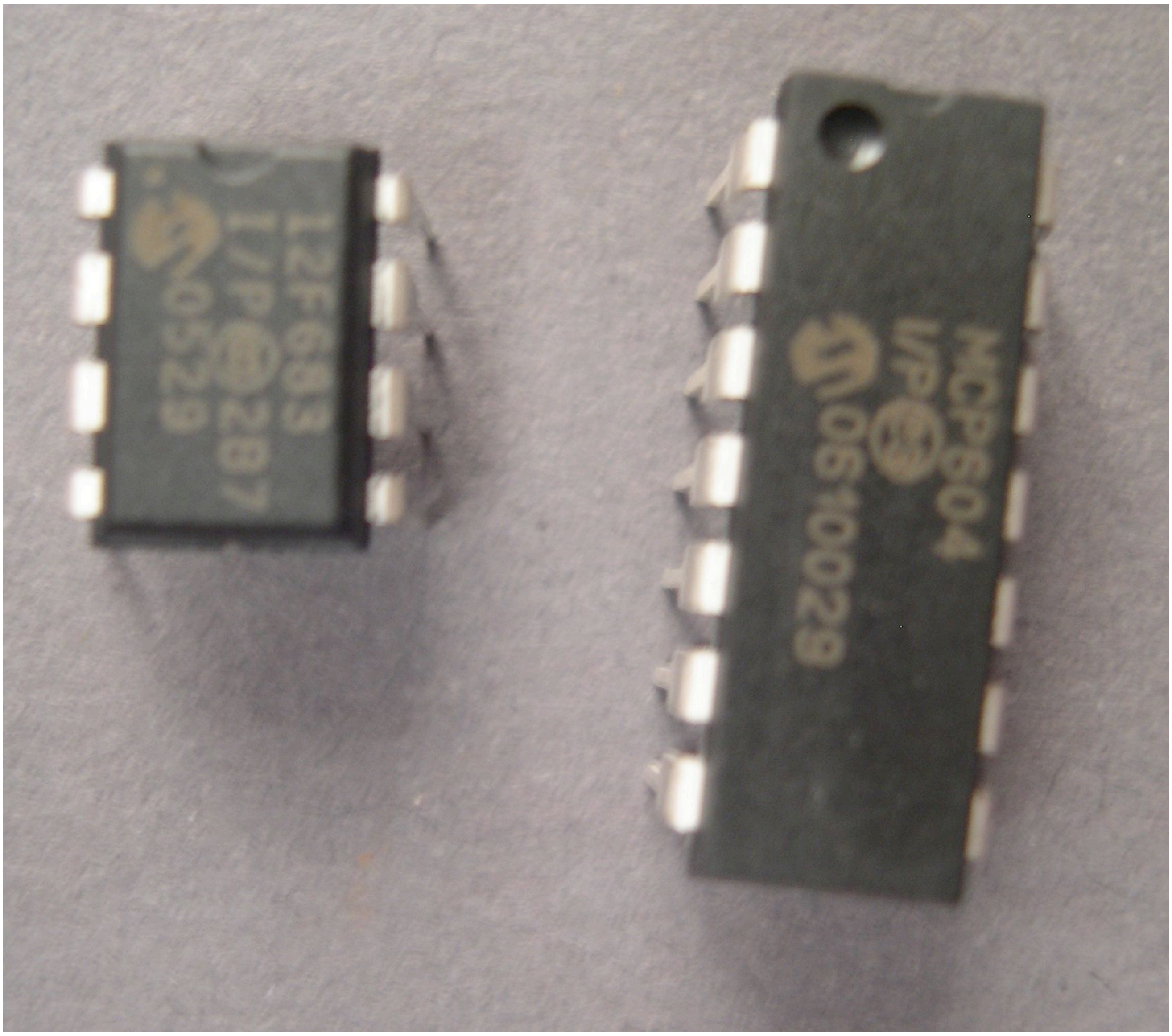

Caps come in many shapes and flavors. A good rule of thumb for the novice is when you’re looking at some circuit or other and you can’t tell what some component is, it’s probably a cap. These particular caps are examples of “film caps” and “ceramic caps”. Both are higher in quality and lower in capacitance (measured in FARADS) than their big electrolytic cousins, and they aren’t polarized, so you can stick them in the board any old way you want.

IC Sockets:
------------



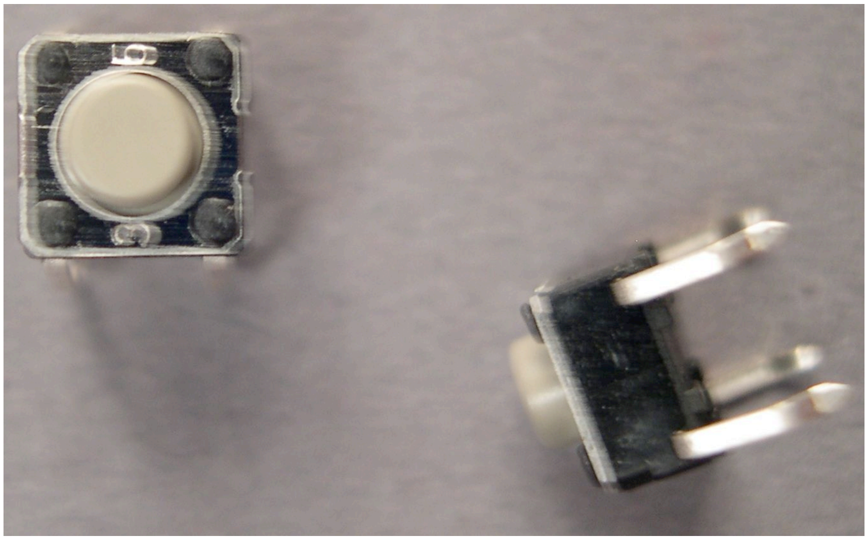

These guys are little solder-in grabbers which hold an IC (integrated circuit, or chip) with spring loaded metal contacts. The 8-pin sockets are made of gold, like an old grill used to be before Paul Wall had the internet going nuts. They're useful if you ever think you might need to change an IC, like if it breaks or you need to reprogram it. Or if you are afraid of messing it up by soldering it all goofyfoot. When inserting these, the notch in the socket goes toward the notch in the silkscreen.

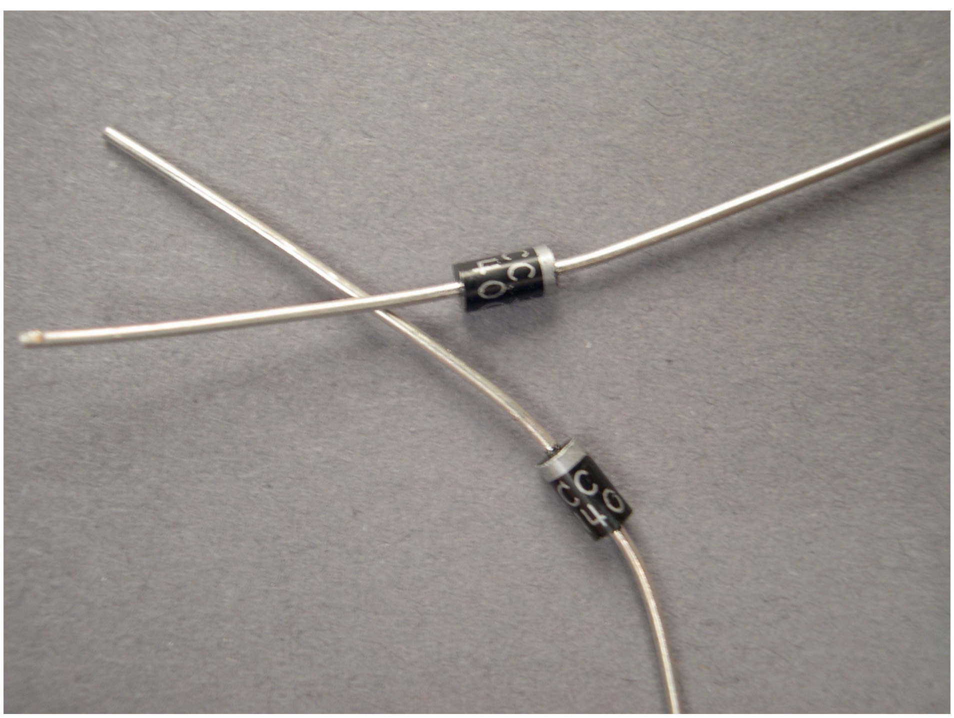
ICs:
------




These guys are the star players of this circuit.  ICs are comprised of as many as millions of transistors on a piece of silicon, all strung together to perform some function or other.  The people who design these things are true wizards. Minor sorcerer’s apprentices like me tend to try and make myself feel better by saying that those people must not have as fun of a life as I do, but it turns out they actually all run coke across the border in homemade hovercrafts in their spare time and date supermodels.

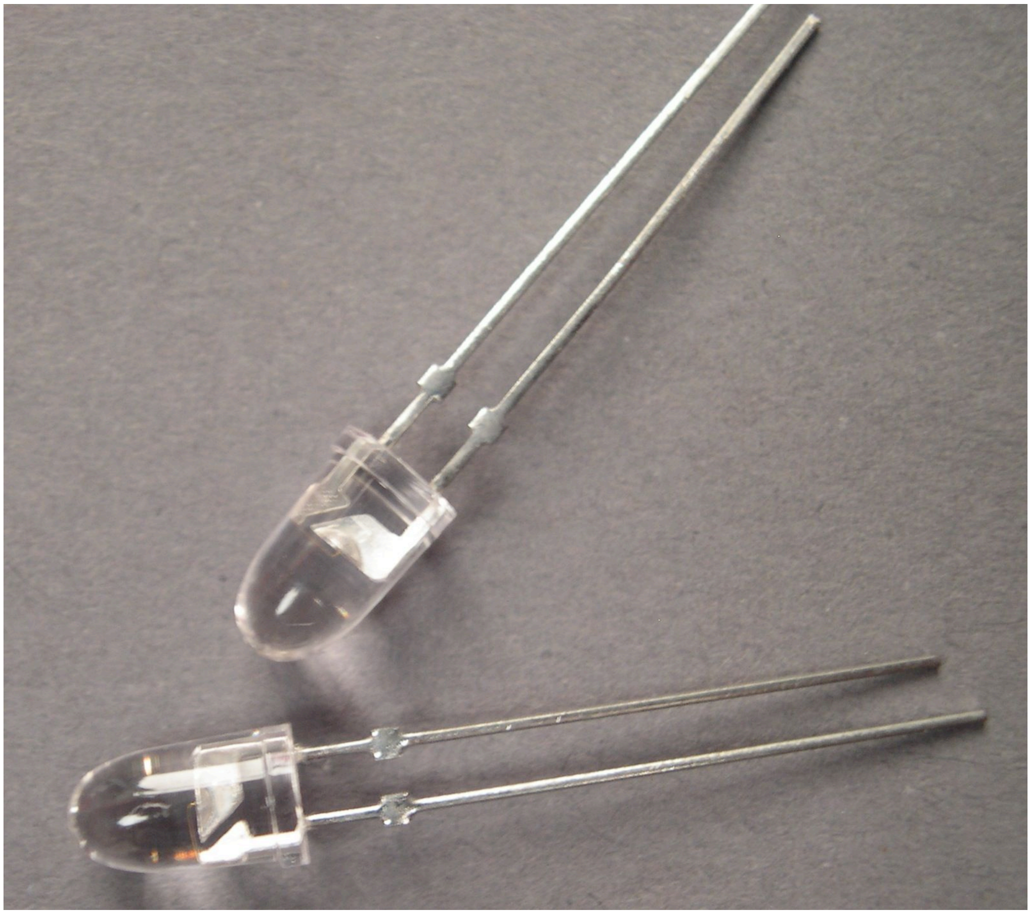
The notch goes up.  The dot on the microcontroller (the 8-pin jobby on the left) and the divot on the op-amp (the 14-pin gentleman on the right) indicate pin one.  UNDER PAIN OF DEATH do not insert these upside down or they will unceremoniously and irrevocably break when you turn the power on.  Be careful not to bend the pins, either.  And wipe your feet.

Switch:
---------

An unexciting but necessary component for user interface.  These are called "tact switches" I guess because you can feel them.  Switches either make a connection or break one when you press the button.  In this case, they make one (they are "Normally Open") and they only make it while the button is held down (they are "Momentary").

Diodes:
--------

Diodes are one-way gates for current, and as such are polar.  In this kit, the diodes handle protecting the inputs from overvoltages and protecting the kit

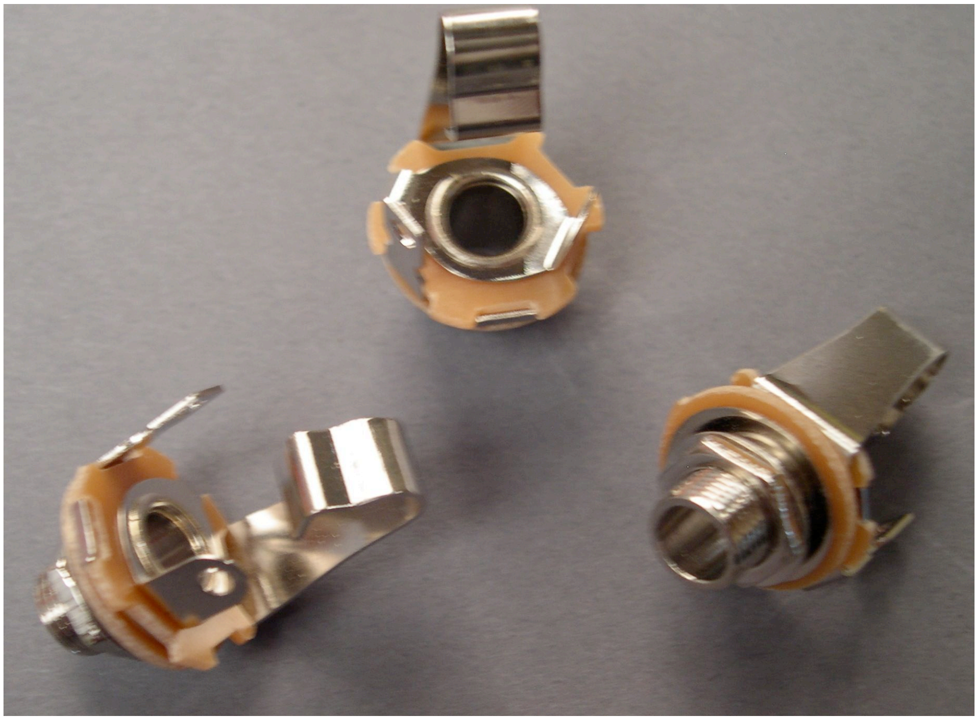
from the batteries being in backwards. Current only flows from the Anode (the positive side) to the Cathode (the negative one). The silver stripe marks the cathode – almost all diodes indicate polarity with a stripe like this. This stripe should line up with the one on the board, under pain of the synth not working and your batteries getting hot / dead.

LEDs:
------



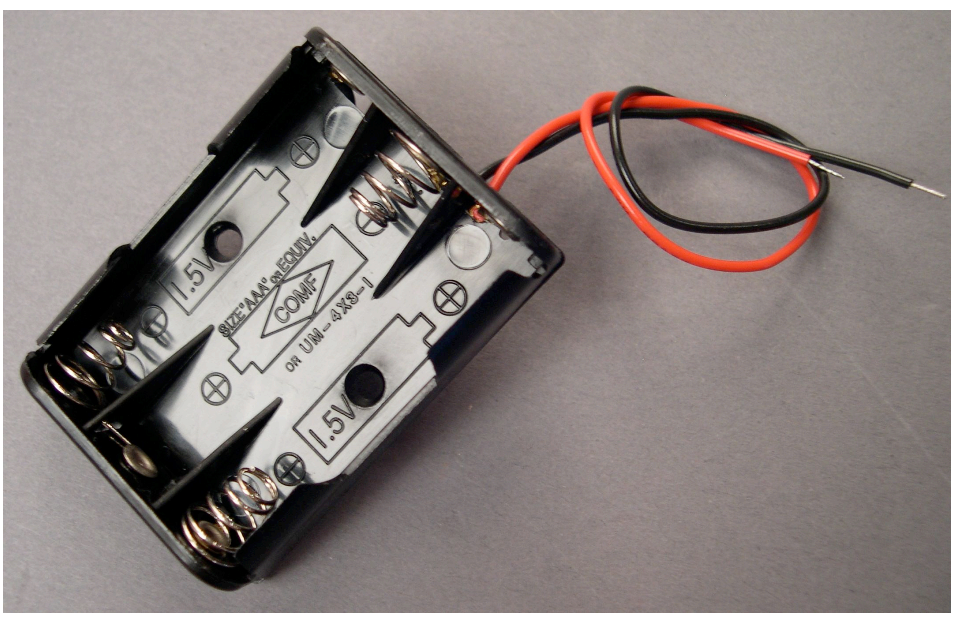
LED stands for "light-emitting diode" and as such is a special case of diode. LEDs are not robust like the 1n400x series of diodes on the last page, but they are a whole lot prettier. They make circuits look good and they can give you feedback about what the circuit is thinking. When current flows the right way through an LED it excites electrons in some exotic physical compounds with cool names like "Gallium Arsenide" or "Indium" and these excited electrons then give that energy back off as light of a certain color, based on the specific chemistry. The long lead of the LED is the anode (+).

Jacks:
-------



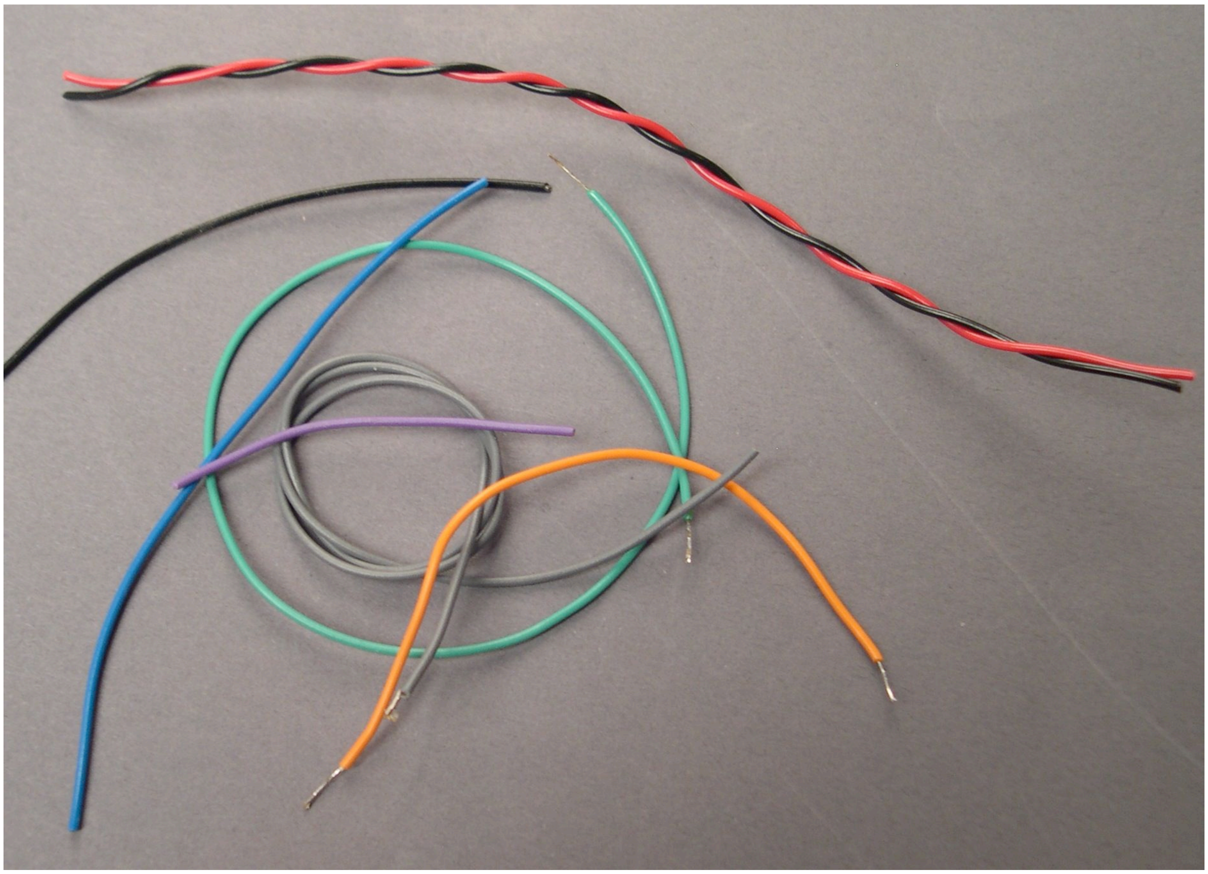
Full-time jack moves like these are used to get signals on and off a PCB and into a cable. Why you'd want them there I couldn't say. These are mono, non-shorting, non-switched, 1/4" jacks. Remember that to preserve the polarity of your signal, the sleeve contact should be connected to the circuit ground on both jacks, but it doesn't really matter.

Battery Holder:
----------------



Holds the batteries. I’m trying very hard to think of something smart to say about this part, and coming up challenged. How about this: “Put the batteries in the same way as in the picture!” or “The red lead is positive (connects to V+) and the black lead is ground,” or “The battery holder comes pre drilled for 4-40 screws, but make sure those screws have a countersink!”

Wire:
------



All wire will work with this kit, but some will work better than others. In general, for hooking things up to a PCB, stranded wire is WAY better than solid wire. Solid wire will break if you bend it a lot. A wire’s diameter is measured in “Gauge” or AWG. The holes on the PCB for the wire connections were made with 28 AWG in mind, although anything around that size will be fine. Also, consider a way to strain relieve your connections to the board and jacks, because if you don’t and just carry ol’ boy around in your fanny pack all the time, the wires will stress and break at the solder joints.

NB: POLARITY MATTERS!
-----------------------

Polarity means: “it only works if it goes in one way, even if it fits in more than one way”. Essentially it means that the component only wants to function in the circuit "in one direction". Usually this is because the current only wants to flow one way through it, or because the voltage across it must remain "in the same direction".

If you solder a polar component in backwards, something bad will happen. I have a bet that most of the problems people will have with this kit will be because they put something in backwards. Double and triple check that you get it right before you solder a component in -- it'll avoid trouble and it’s a lot easier to solder than desolder. Fortunately it isn't that hard. The printed circuit board has markings that indicate the correct way to put a polar component in.

*The diodes have little bars on the parts that match up with the little bars on the PCB where they go.

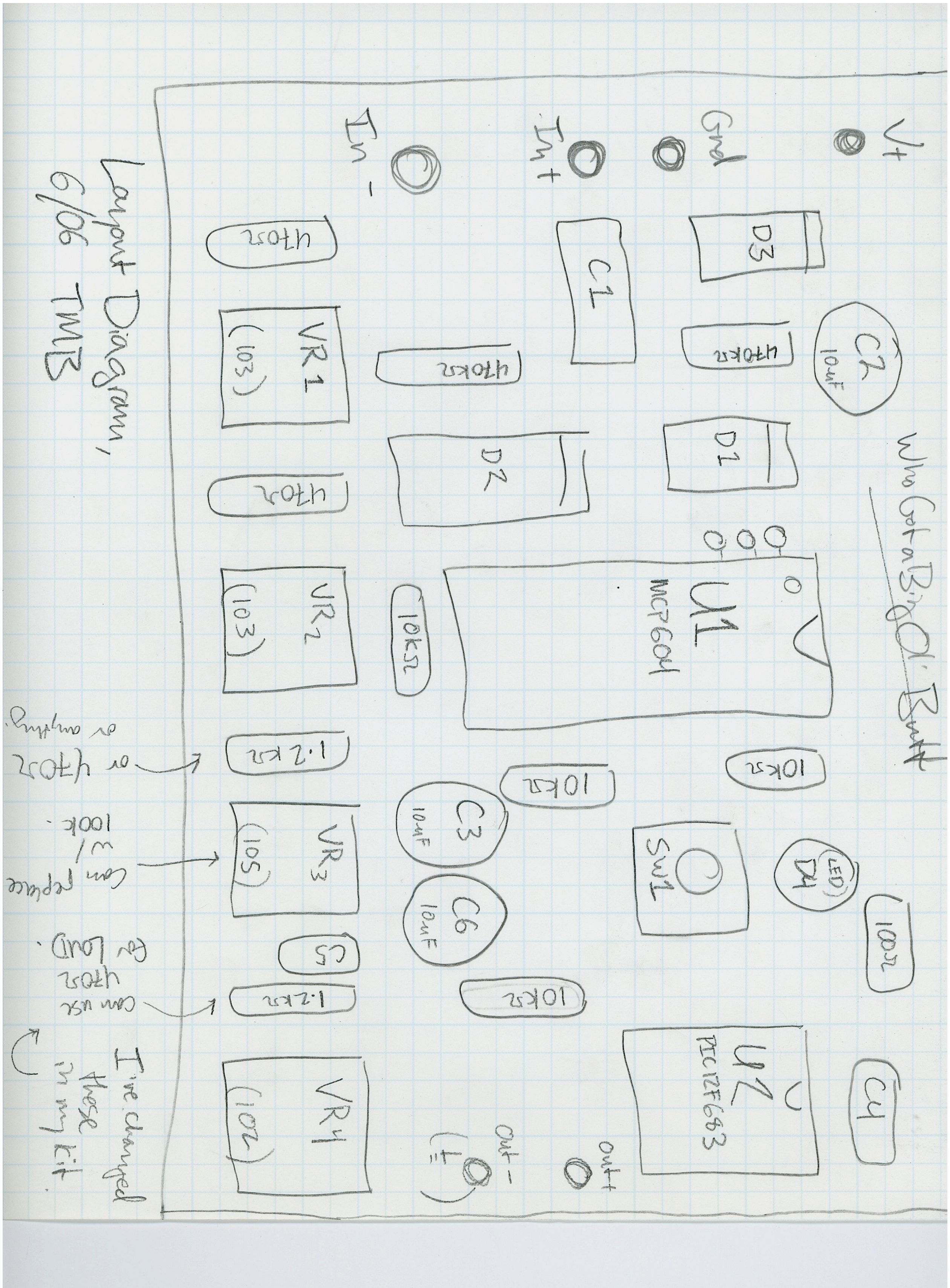
*The big electrolytic polar caps have a gray stripe on one side which indicates the NEGATIVE side. On the board, there's a + sign which indicates the POSITIVE side. So the stripe on the cap goes away from that side.

*The LED has a flat side which should match the flat side on the PCB silkscreen.

*The ICs (the chips, the black bug things) have a little divot taken out of one of the short sides of the chip. Match that up with the divot on their drawing on the board. They also have a little dot above one of the pins. This dot should always be in the upper left corner if you're looking down at the board.

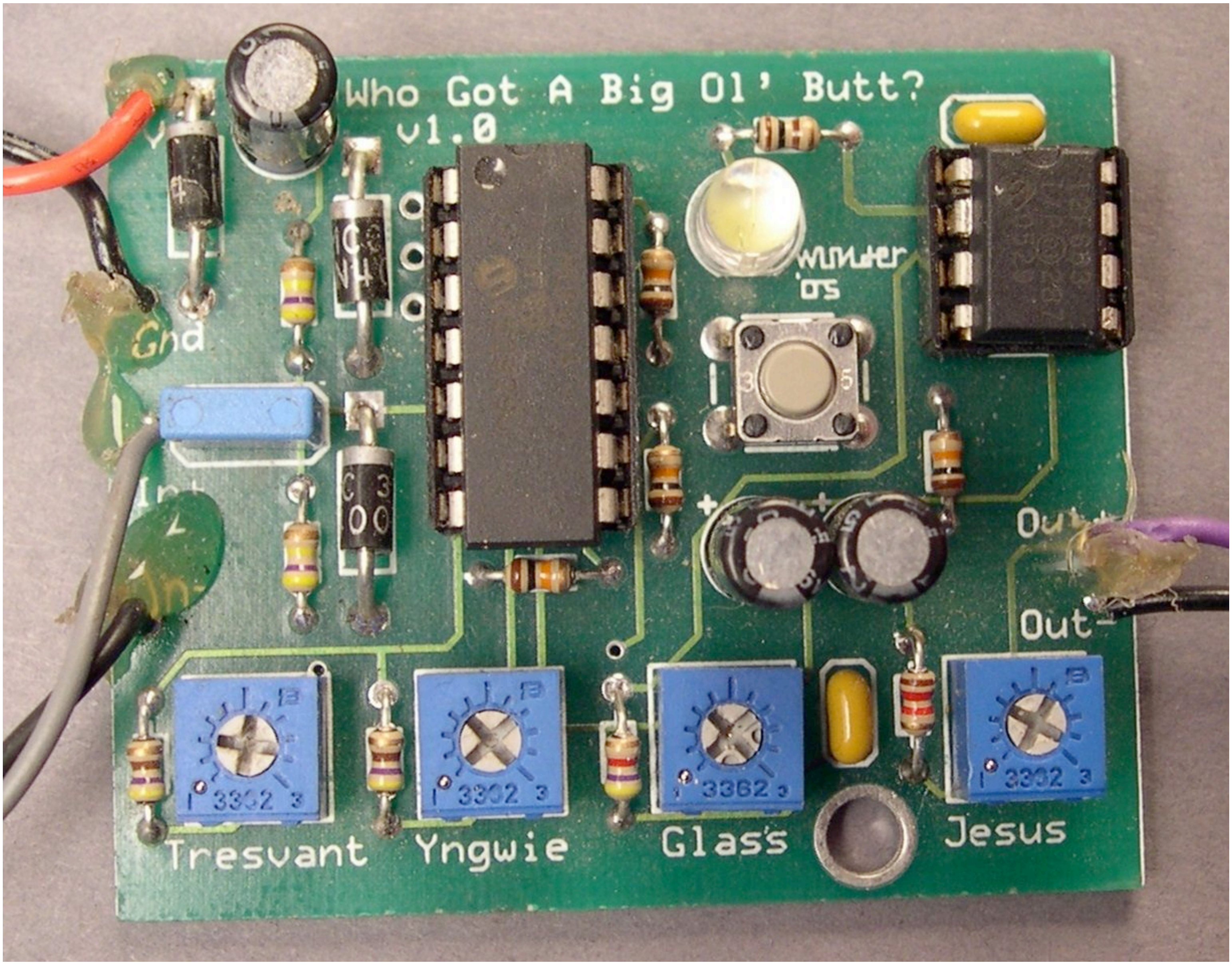
## 5a.) Where does all this stuff go?

Here's a layout diagram I drew with all the component placements. I can't really draw, forgive me, but this is where all the parts belong on the board:

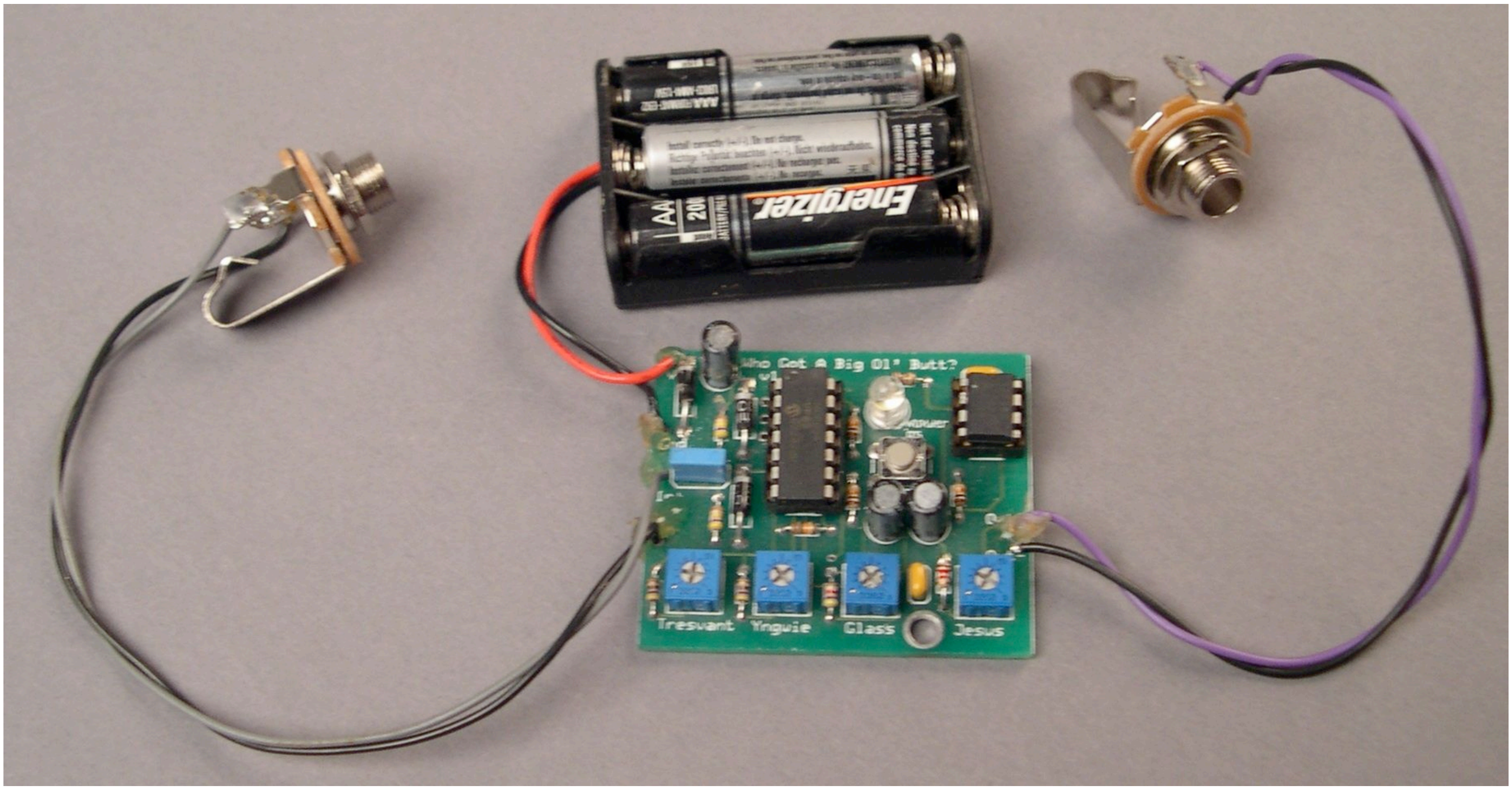

Layout Diagram, 6/06 TMB
Who Got a Big Ol' Butt
V+
Gnd
In+
In-
Out+
Out-
D1
D2
D3
D4 (LED)
C1
C2 10uF
C3 10uF
C4
C5
C6 10uF
U1 MCP604
U2 PIC12F683
SW1
VR1 (103)
VR2 (103)
VR3 (105)
VR4 (102)
470kΩ
470kΩ
470Ω
470Ω
10kΩ
10kΩ
10kΩ
10kΩ
10kΩ
100Ω
1.2kΩ
1.2kΩ
or 470Ω or anything.
can replace w/ 100k.
can use 470Ω & LOUD.
I've changed these in my kit.

HERE'S WHAT THE KIT SHOULD LOOK LIKE ONCE YOU'VE SOLDERED IT UP:
-----------------------------------------------------------------

Use this as a guideline. If you get lost, refer to this drawing. Yours should look like this when it's done:



## HERE'S WHAT IT LOOKS LIKE WITH THE BATTERY HOLDER AND JACKS ATTACHED:



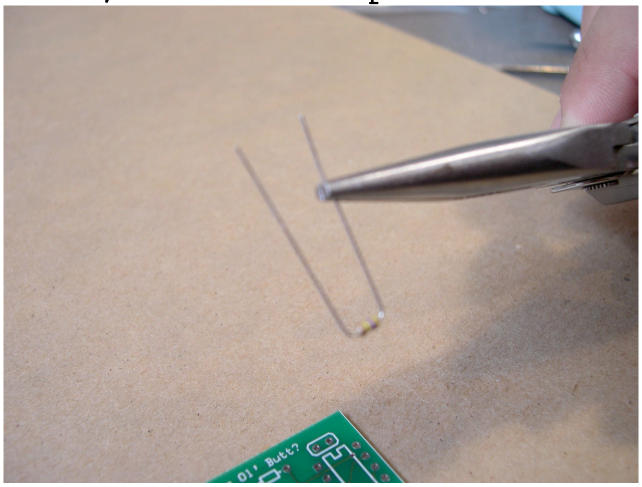

## 5c.) Soldering Our First Component:

What follows is a little tutorial on how to solder. I made a short movie about how to solder which will probably help you at least as much as this tutorial, and it should be online somewhere near where you found this. Check it out if you have the time and inclination. Also, If you have the Ghazala book, read his section on soldering -- it's really good.

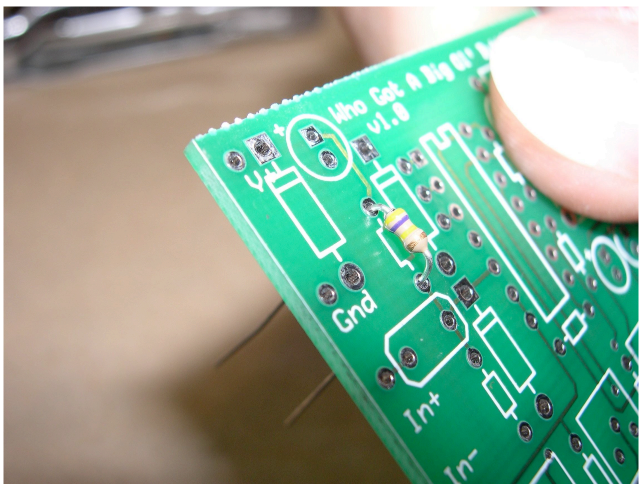
To begin:

We'll pick a resistor, because they're very forgiving of heat and abuse. If you can find some other poor piece of electronics to practice soldering on a few times, your kit will probably turn out a little better -- but if not don't worry. Soldering isn't that hard. You'll be a pro by the end of this project. If you're already a wizard with solder, don't worry about reading this section.

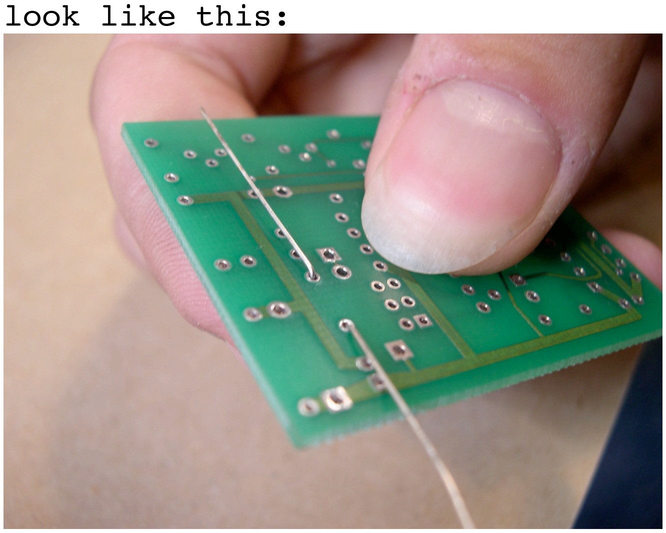
First, bend the component leads so they look like this:



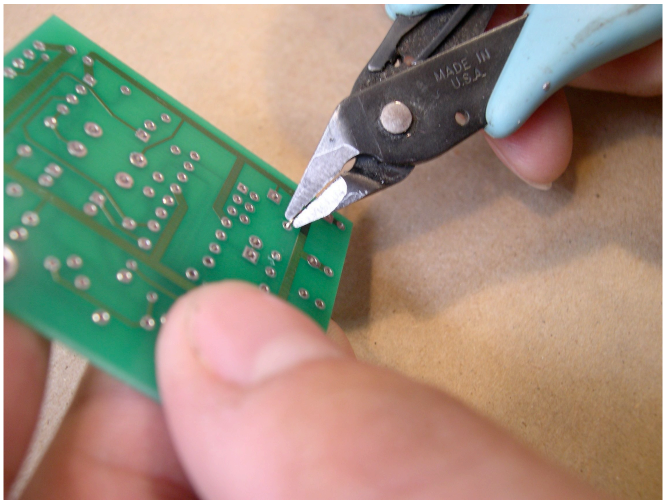
Now, insert the resistor in its proper spot on the board.  Push it all the way down until it looks like this:



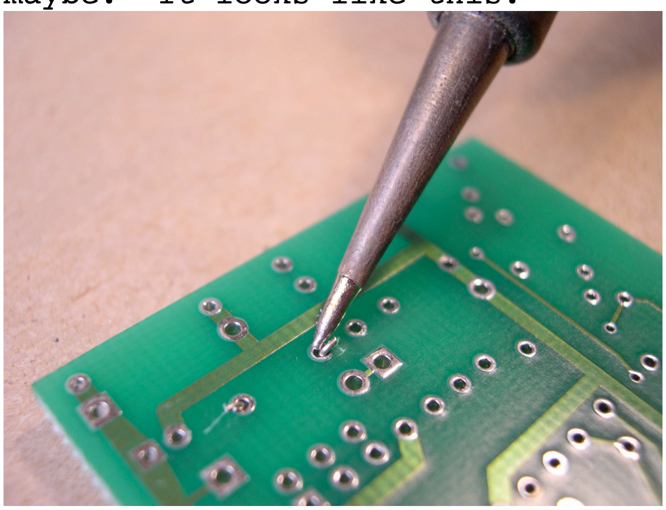

Keeping your finger on top of the resistor so that it doesn't fall out, turn the board over and bend the leads so that the resistor can't fall out.  It should look like this:



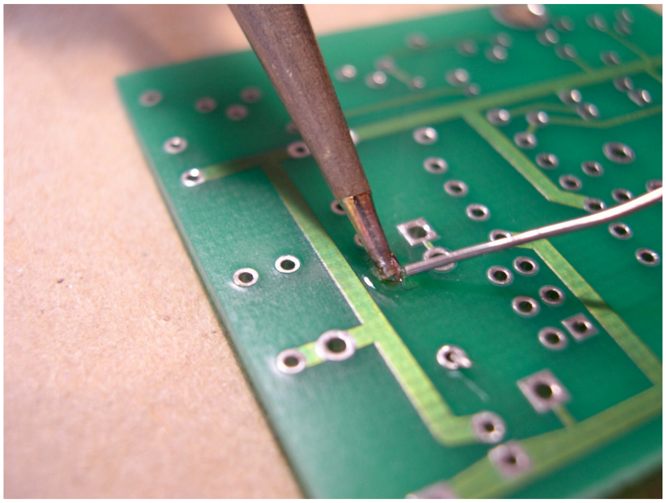
Snip off the excess lead length with your diagonal cutters so that they look like this:



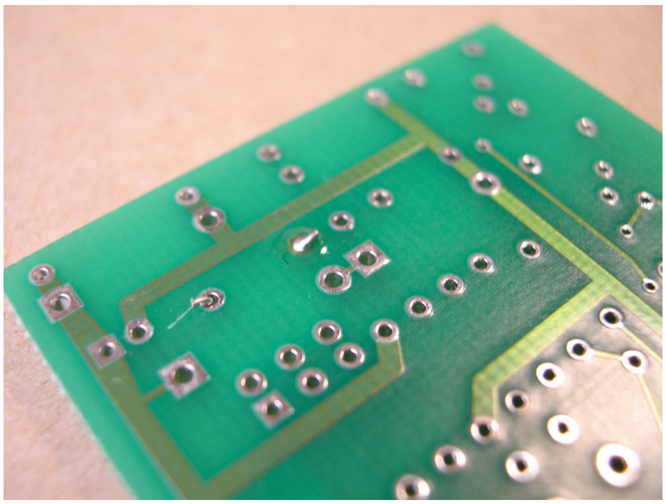

Now comes the fun part -- soldering.  Make sure your iron is hot and that you've cleaned it by wiping the tip across your wet sponge (if you got one).
First, with your good hand hold your iron against the component lead and the pad to which you are soldering.  You should only need to do this for two seconds, maybe.  It looks like this:



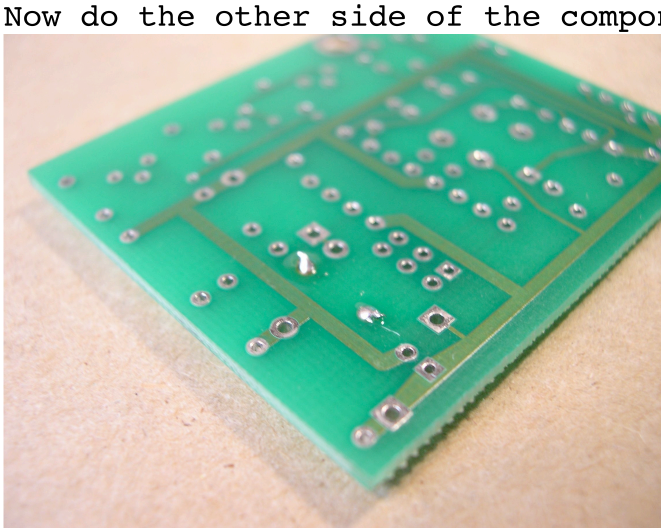
Next, with your other hand, apply solder to the junction of the component lead and pad.  Here are some action shots:



Once you see the solder flow over everything, take the iron and solid solder off the connection point. The molten solder will harden almost immediately. It looks like this:



Now do the other side of the component. The finished component looks like this:



Now move on, young wizard, and solder all the rest of the components onto the board! But here are some rules to keep in mind about soldering:

1.) When cutting bending leads and soldering, it is often easier to insert the component, bend the leads over AND THEN SOLDER, cutting the excess lead after the solder joint has been made.  This isn't as good, because you'll usually cut into the solder meniscus and the joint won't be as perfect, but it is sometimes WAY easier.  Do this if you have to.

2.) If you find yourself keeping your iron on the board / part for more than 5 to 10 seconds, something is wrong.  Take the iron away and let the component and board cool down before you try again.

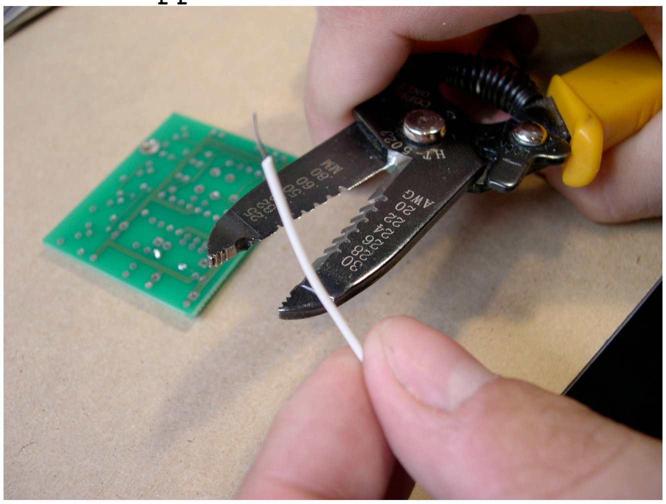
3). The sockets will probably be the hardest thing for you to solder because you can't bend the leads.  You'll figure it out, but don't burn your fingers.  You won't need to trim the socket leads, either.

4.) If you mess up and bridge some contacts with solder, never fear.  Remove the excess solder with desoldering braid, available from Radio Shack and many mail order component places.

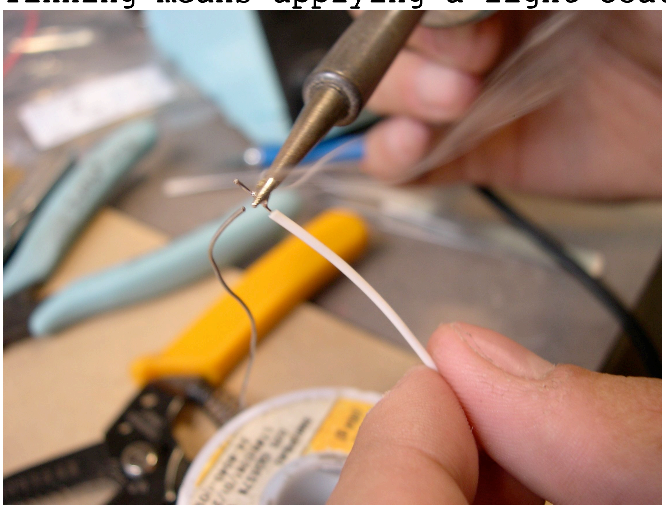
5.) DON'T EAT SOLDER.  It's mostly lead and therefore very bad for you.  And wash your hands when you're done at your workbench.  Seriously.  And try to solder in a place with adequate ventilation.  Making one kit won't kill you no matter what you do, but now is a good time to develop good habits.

Installing Jacks, Tinning Wires:
--------------------------------
Soldering wires works like this.  First strip the end:

The stripped end will look like this:



Then "tin" the wire.  This will make soldering to a board or jack WAY easier. Tinning means applying a light coating of solder to the exposed wire:



Then just solder it like a regular component lead.

Battery Holder:
---------------
The battery holder leads solder in just like any other wire.  However, you must make sure to solder the batter holder leads into the right places on the board. The red lead (Battery +) goes to the pad labeled "V+" and the black wire (Battery -) goes the one labeled "Gnd".  Remember also that WGABOB doesn't have a power switch right out of the box, so if you put the batteries in, the thing will turn on.  So don't do that until you're ready to turn the device on.

Socketing the chips:
--------------------
Once everything else is done, put the chips into the sockets. DON'T SOLDER THE CHIPS IN. They don't take abusive soldering as well as the passive components, and moreover, if something in the kit is messed up they are the most likely things to break. Replacing a chip once it's been soldered in is MUCH HARDER than just pulling it from the socket and putting in a new one. Plus, the microcontroller can have new programs entered into it if you can pull it easily.

MAKE SURE YOU PUT THE CHIPS IN THE RIGHT WAY. With the divot closest to the "Who Got A Big Ol' Butt?"

When pressing them in, be careful. For some reason, the chips come with their leads bent outward a little bit. You may need to squeeze the two rows of pins together a little bit to get the chip to fit in the socket. DON'T RUSH. If you push the chip into the socket without everything lined up right, you'll bend the pins on the chip. If you do this, pull the chip back out and CAREFULLY straighten the pins. If you do this more than a couple times, you'll break a pin and need a new chip.

Assembly tips:
---------------
Don't put in anything backwards and don't burn anything up. When you think you've finished, double check that all your components are there and in correctly. Then triple check. Really. If you turn the thing on and something gets hot right away, turn it off and do some more checking. If you're fast and lucky, you may have avoided tragedy.

I have confidence in you. You'll work it all out -- and you'll feel really good when you do. I know I did.

-----------------------------
5d.) Troubleshooting Your Kit
-----------------------------

Man, there are a million ways to break electronics. Here are some symptoms I can think of happening, and what I'd check first if I saw them. I'm just guessing, though.

Symptom: Nothing works. The batteries are really hot and run out really fast.

Problem / Solution: This is bad. This means there's a short somewhere on your board. Make sure your batteries are connected to the synth right (i.e., not backwards) and that diode D3 is in the board the right way. After that, make sure your ICs are in the right way. After that make sure there aren't any solder bridges that you accidentally made and shouldn't have.

-

Symptom: Nothing works or everything seems weird and one of the components is getting really hot.

Problem / Solution: Turn the synth off! When things get hot it means that they are conducting a lot of current, which in this synth is not normal. Usually it means a short. The hot component is either busted or on its way to being busted. Again, make sure that the polar components are in the right way. Make sure that one of the outputs of one of the chips isn't inadvertently tied to a supply rail or another output.

-

Symptom: I turned it on and nothing got hot, but nothing blinks and I can't hear the output.

Problem / Solution: All your batteries and chips are present, right? Make sure that the board is getting power by double checking the battery connections. Make sure your chips are socketed correctly and that everything is soldered in.

-

Symptom: I can hear the output, but the light doesn't do anything.

Problem / Solution: The LED is probably in backwards. It won't break, but it won't run, either. Turn it around and put it in the other way.

-

Symptom: I can see the light do something but there's no sound at the output.

Problem / Solution: Are your volume knobs up? Are your jacks soldered on correctly? Is everything on the board that should be?

-

Symptom: It's a pain in the ass to adjust the trimmers!

Problem / Solution: It sure is, isn't it! Solder on some real potentiometers and build a chassis. That's what I did to mine. DIY, my friends. Call me if you need help. The reason this kit is so cheap is because I've left it up to you to rock your kit out.

-

Symptom: The kit sounds cool but can't it do (X)?

Problem / Solution:  It probably can.  Maybe.  If there's something you want the kit to do, see if you can figure it out.  If you can't, let me know and I'll either point you in the right direction or write a new program.  That's what's so exciting about a digital kit like this one!  You can get a whole new instrument just by putting in a new chip.  And a lot of homework.

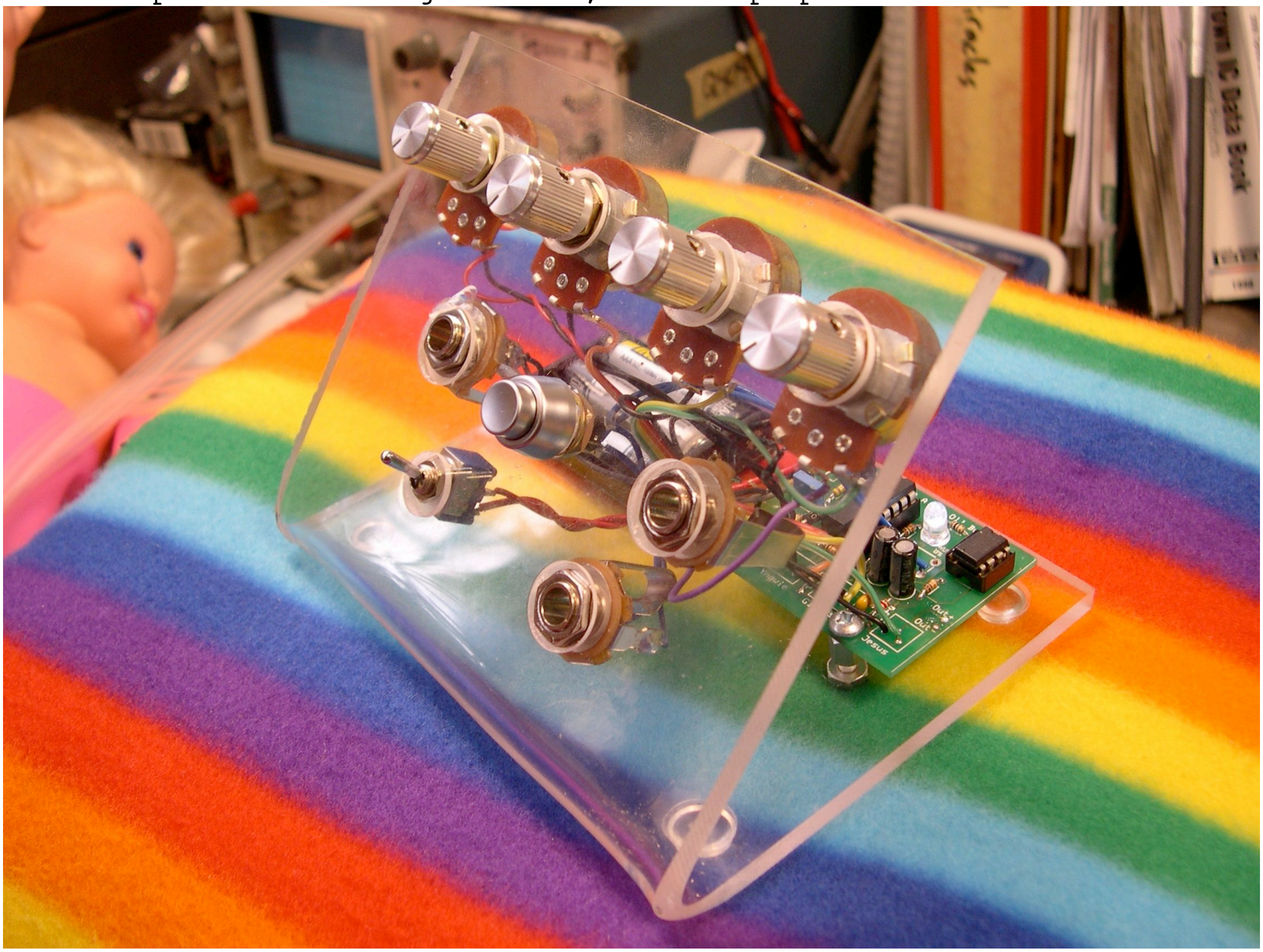
---------------------------
## 6.) How to Play the Synth
---------------------------

The "instructions" are very simple.  One tap of the button will toggle the PIC's output off and on, and holding the button down for a second or two will change to the next program.  There are five programs at the time I'm sitting here writing this.  The knobs are the following:  "Tresvant" controls the sensitivity of the PIC to an input signal.  "Yngwie" controls the output level of the un-synthesized input.  "Glass" controls the level of the output from the PIC -- the weird synth output level.  "Jesus" is the master volume.

After that, let your freak flag fly.

-----------------------------------------------------------
7.) Suggested Mods, Circuitbending Instructions, Ephemera
-----------------------------------------------------------

Here's a picture of MY Big Ol' Butt, which I pimped out last week:



I made a chassis out of acrylic sheet and mounted real panel mount potentiometers with aluminum knobs on it for easier adjustment than the trimmers. The switch is now a real button, and the third jack at the bottom right of the control panel is in parallel with the switch and goes to a footpedal which I use in addition to the panel mounted button to change programs and turn the device on and off. The little toggle switch is a power switch. Some of the component values are different stuff I had around my bench, but all of this is totally doable. I did it in a day. The parts and materials to do all of this stuff were probably in the neighborhood of $15, and most of that cost is those knobs. Now I kind of want a switch to turn off the feed-through from the original instrument input, so that the audio out is ALL generated by the microcontroller. You’ll get little bees in your bonnet, too, if you fiddle with your kit enough.

Oh, and one of the reasons I picked the parts I did for this kit is so that you could use it with a relatively wide supply range. You can run it with the battery pack I included, but it's even happier running off of 5v. Alternatively, it works fine at 3v -- build a supply for this device if you want or run it off a wall wart.

Absolute Maximum Ratings (V+):
-0.5 - 7vDC
Recommended Operating Range:
2.7-5.5vDC.
NOTE: THIS MEANS IT WON'T WORK WITH A 9v BATTERY. It'll fry.

If you want to circuitbend your synth, read the Ghazala book. My humble suggestions: get it working right first, and THEN circuitbend it. Don't mess with tying anything directly to the battery rails unless you know what you're doing.

If you want to learn how to do this on your own, Blessed Be Thy Name!

It's hard to learn DIY electronics design -- but it is arguably the most rewarding thing I've ever done. Get all the books in the suggested reading list and start building. Get a breadboard, a multimeter, a scope, and a power supply. Get the free catalogs from Digikey and Mouser. Literature wise: The projects from the Mimms and Anderton books are easy, but the real magic is in Horowitz and Hill. Those guys are geniuses, but they assume a lot of discipline and work on the part of the reader. Once you've got that licked, go to Gray and Meyer's "Analysis and Design of Analog Integrated Circuits", and if you understand that, call me and tell me what it means.

Seriously, once you've got this thing built and got some confidence under your belt and you want to keep going, drop me a line and I'll totally talk your ear off. The more circuit nerds in this world, the better.

-----------------------------------------------------
## 8.) Supplemental Datasheets / Recommended Reading
-----------------------------------------------------

Most important to this kit are the datasheets for the MCP604 Op Amp and the PIC12F683 microcontroller:

The op amp:
http://ww1.microchip.com/downloads/en/DeviceDoc/21314f.pdf

The PIC:
http://ww1.microchip.com/downloads/en/DeviceDoc/41211B.pdf

And a useful datasheet that might teach you a little about op-amps and give you an idea of what to do with U1d:
http://cache.national.com/ds/LM/LM124.pdf

And here are some useful books, organized by Apprentice / Wizard:

Apprentice:
-----------
* Circuit-Bending: Build Your Own Alien Instruments (ExtremeTech) by Reed Ghazala
* Getting Started in Electronics by Forrest M. Mims III
* Timer, Op Amp, and Optoelectronic Circuits & Projects by Forrest M. Mims III
* Handmade Electronic Music: The Art of Hardware Hacking by Nicolas Collins
* Building Power Supplies by David Lines
* Electronic Projects for Musicians by Craig Anderton

Wizard:
-------
* The Art of Electronics by Paul Horowitz and Winfield Hill
(This is the best book about electronics, ever, and my favorite.)

* Linear Applications Handbook, National Semiconductor.  1973.

* The Art and Science of Analog Circuit Design, Ed: Jim Williams.

There are lots of other more specific books in my library (the wizard books in particular get pretty deep into their specific fields of arcana) and the internet is full of useful information.  If there's anything you want to know more about, and you can't find a good resource, just drop me a line.  Also, I'd love to hear about what you're doing with your kit.

The final page of this manual is the resistor color code section from Horowitz and Hill.  I like to call it "Appendix A".

Happy Building!
TMB
Tm_bailey@hotmail.com
(312) 307-9296

# APPENDIX C

# THE 5% RESISTOR COLOR CODE

Low-power axial-lead carbon-composition and film resistors with 2% to 20% tolerances have a standard set of values and a standard color-band marking scheme. Although it may seem diabolical to the beginner, the practice of color banding makes it easy to recognize resistor values in a circuit or parts bin, without having to search for a printed legend. The standard resistor values are chosen so that adjacent values have relative ratios of about 10% for the 2% and 5% tolerance types and 20% for the 10% and 20% tolerance types. Thus there are many values that could be described by the color code but that are not available.

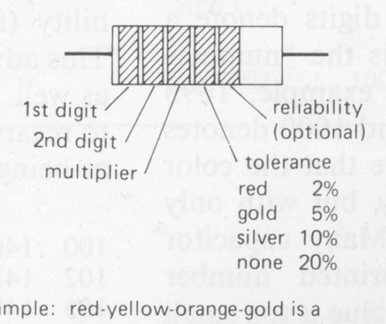
Two digits and a multiplier digit determine the resistor value, and resistors are color-banded in that order starting from one end of the resistor (Fig. C1). A fourth tolerance band is usually present, and occasionally you'll see a fifth band for other parameters (such as yellow or orange band for MIL spec reliability rating).

Here is the set of standard values for the first two digits (lightface type indicates 2% and 5% only):

| | | | | |
|---|---|---|---|---|
| **10** | 16 | **27** | 43 | **68** |
| 11 | **18** | 30 | **47** | 75 |
| **12** | 20 | **33** | 51 | **82** |
| 13 | **22** | 36 | **56** | 91 |
| **15** | 24 | **39** | 62 | **100** |

Carbon-composition resistors range in price from 3 cents each (in quantities of 1000) to 15 cents (quantities of 25). Distributors may be unwilling to sell less than 25 to 50 pieces of one value; thus an assortment box (made by Stackpole or Ohmite) may be a wise purchase.

| digit | color | multiplier | number of zeros |
|---|---|---|---|
| | silver | 0.01 | −2 |
| | gold | 0.1 | −1 |
| 0 | black | 1 | 0 |
| 1 | brown | 10 | 1 |
| 2 | red | 100 | 2 |
| 3 | orange | 1k | 3 |
| 4 | yellow | 10k | 4 |
| 5 | green | 100k | 5 |
| 6 | blue | 1 M | 6 |
| 7 | violet | 10 M | 7 |
| 8 | gray | | |
| 9 | white | | |




example: red-yellow-orange-gold is a 2, 4, and 3 zeros, or 24k 5%, resistor

Figure C1