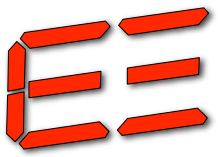
Peggy 2.0

Light Emitting Pegboard Kit

An open-source hardware+software project designed by



Evil Mad Scientist Laboratories

Making the World a Better Place, One Evil Mad Scientist at a Time

Support: http://www.evilmadscientist.com/forum/

Distributed by
Evil Mad Science LLC
http://evilmadscience.com/

Evil Mad Science

Kit version 2.3
Manual v. 2.30

Intro + Tour

Hi, I'm Peggy!

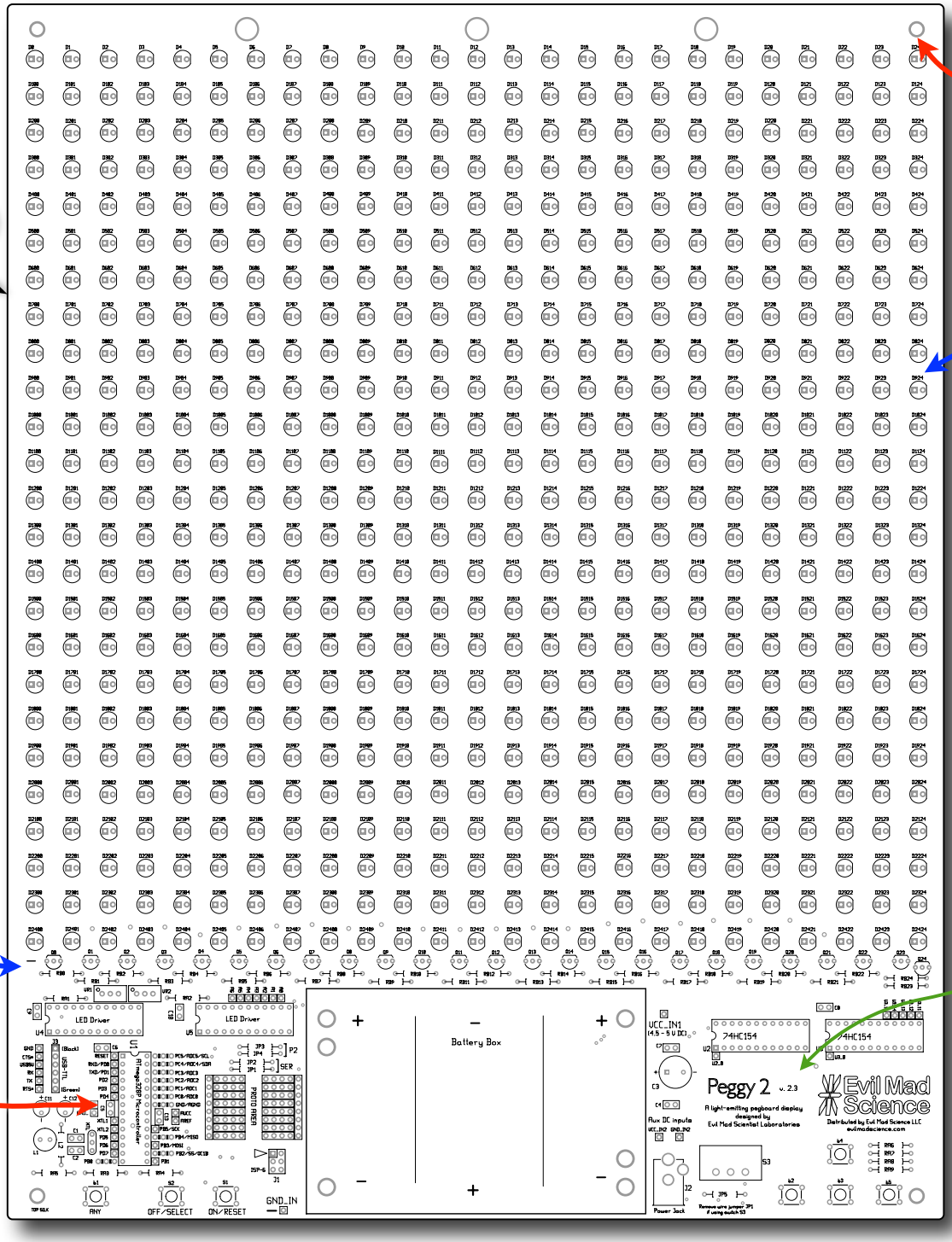
Peggy 2 is a tough and versatile LED "pegboard" display that can drive a few or a lot of LEDs for almost any purpose. Peggy can run on batteries or external power. Peggy is programmable, open source and hackable. Peggy can be the one to figure out how to drive all your LEDs.

Printed circuit board:
Outline: 11.320" x 14.875"
(About 28.8 x 37.8 cm)
Extra thick for stiffness, too!

Transistors and resistors along bottom edge of LED field.

CPU: ATmega328P, a type of AVR microcontroller

(Also: Programming interfaces, places to put switches, and more.)



Mounting holes: Top center: Hang it on a nail. Top 1/4 & 3/4: hang it on a string.

Corner mounting holes are 0.141" in diameter and are located 1/4x1/4" from each of the corners.

LED Field: Peggy can fit up to 625 of your favorite LEDs in a big square grid. Each LED location has room for a 10 mm LED, although smaller 5 mm (T-1 3/4, "standard size") and 3 mm LEDs will work just as well.

Each LED location is numbered by its row and column location.

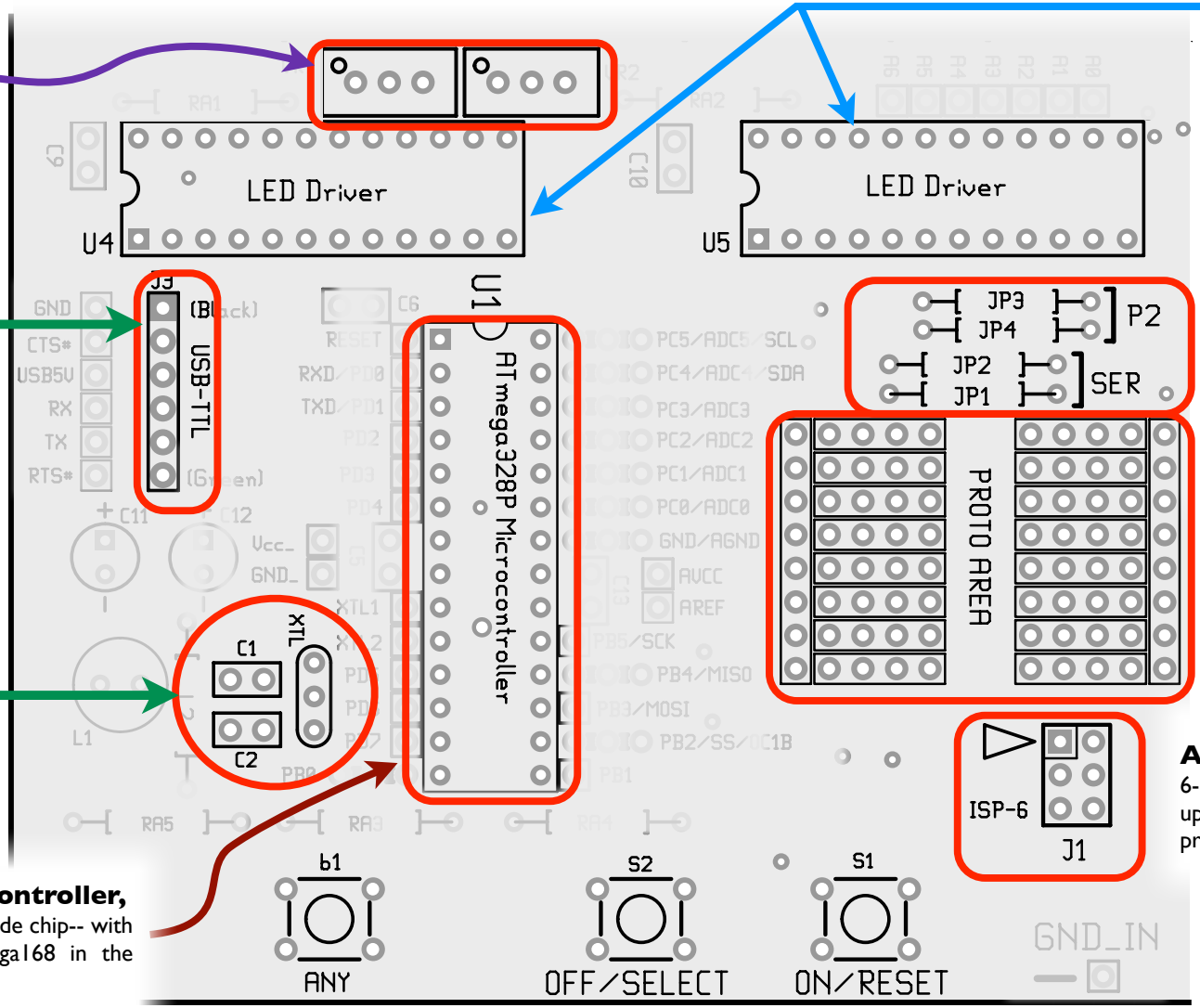

(US Quarter for scale)

Peggy 2 circuit board version 2.3: (Several improvements since the first Peggy 2.)

Bottom center and right: power management, optional button locations, and chips that peggy uses to address the LEDs.

U4 and U5: STP16DP05 LED driver chips.
The microcontroller uses these to control the *columns* of the display. (Over what's called an "SPI" interface.)

Brightness control:
Two trimmers to adjust brightness of left and right sides of display.



USB-TTL interface
If you have a USB-TTL cable, you can use this port to program or communicate with Peggy, much like you would with an Arduino board.

Option jumpers
Configure the optional serial port.

Breadboard-style Prototype area!
For you to use in any way that you see fit. All the holes inside a given rectangle are wired together.

Clock area: Crystal oscillator and two little helper capacitors. [Crystal location has three pins, just in case you wanted to use a 3-pin ceramic resonator instead.]

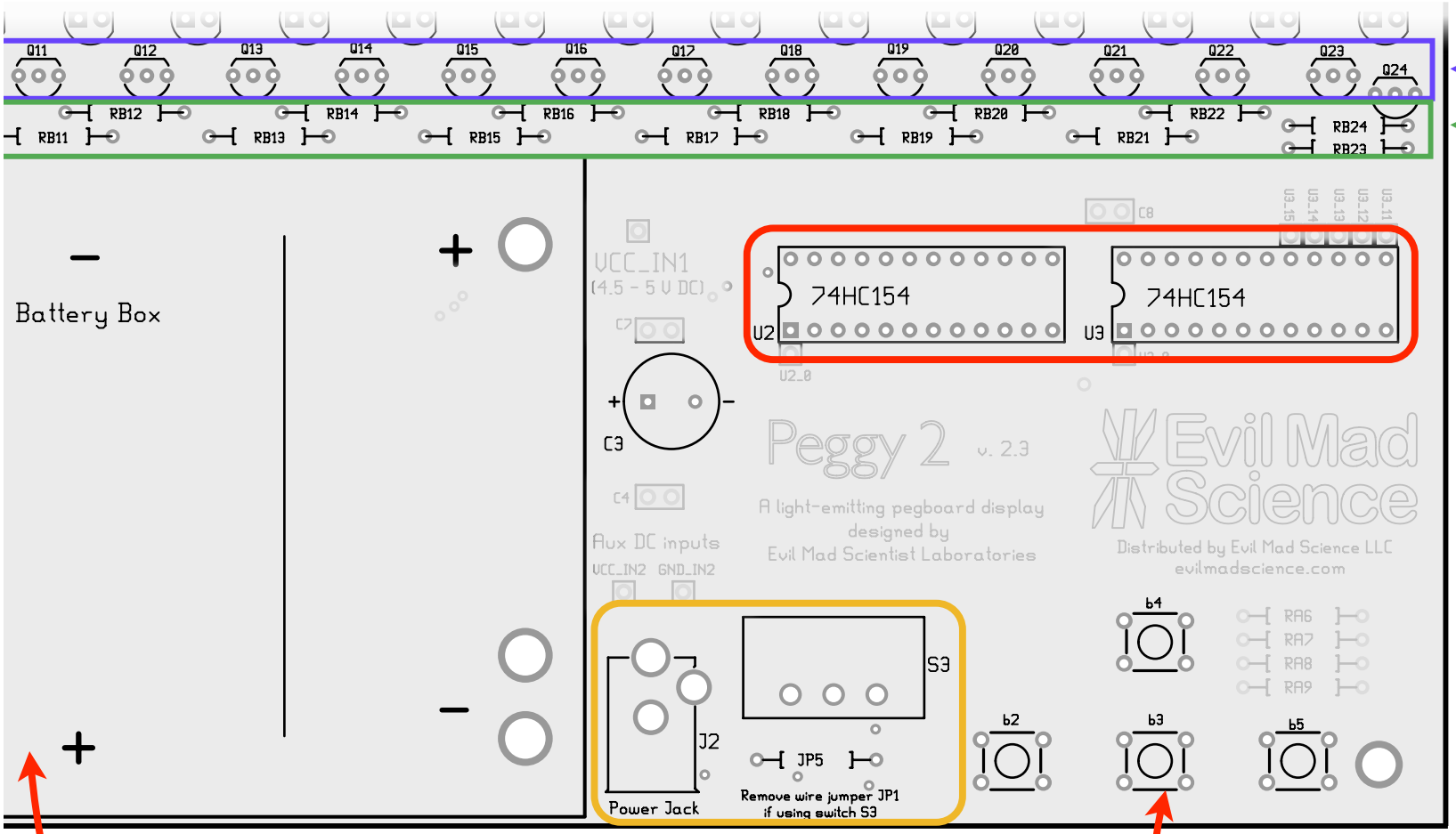
AVR-ISP interface
6-pin connector can hook up to your AVR in-system programmer, if ya got one.

U1: ATmega328P microcontroller,
in a socket. This is a nifty new upgrade chip-- with twice the memory of the ATmega168 in the original Peggy 2.0 kit.

Extra, labeled holes are provided on all of the pins for your hacking pleasure.

Buttons!
Reset and "OFF/SELECT" buttons are standard. "Off" is only a software mode-- only *mostly* off. To save power, use the switch instead.

If reprogramming the display, you may want to add an "Any" key for use in your own programs.



Row of transistors

Row of resistors

U2 and U3: CD74HC154's
"Demultiplexer" helper chips that the microcontroller uses to select and drive the *rows* of the display, through those resistors and transistors.

Battery box goes here

J2, S3: Switch & External power
Power Switch + place for optional power jack.

The switch chooses the power source:
Battery power: On to the *right*
External power jack: On to the *left*.

Extra locations for optional buttons
Not much for these to do if you're building a static LED display, but if you're making something interactive, extra buttons might come in handy....

What do *you* make of this?

Our standard assembly instructions, beginning on the next page, will produce a static "pegboard" display, that will light up LEDs in whichever locations you choose to install them, with current applied to each LED. For optimal uniformity, the display should be fully populated or reprogrammed to light only the LEDs that you want lit.

The LEDs are driven in an energy-efficient multiplexed arrangement. Adding additional functionality, for example, turning on specific LEDs or simple animation or even a true interactive display, is possible by reprogramming the display through one of the two provided interfaces. A third interface, I2C (aka TWI), is available for advanced applications in which it is desired to feed live data to the Peggy2.

Hardware hackers may want to download and flip through the schematics before going further, just to see if inspiration strikes. You can download them here:
http://www.evilmadscientist.com/source/p23schem.pdf

A wide variety of hacks, mods, and improvements are possible; the circuitry was designed with hacking in mind. A prototype area is provided to add extra components and extra access holes have been added to allow direct access to the microcontroller pins. Go for it.

Advanced users *may* want to consider trying out the popular Peggy 2 "serial hack" which is now possible with a simple wiring change. You'll see more about this option on page 12.

If you do wish to reprogram the display...
Two different types of external interface are supported.

Peggy 2 can be programmed through the Arduino software environment (www.arduino.cc), using an FTDI USB-TTL serial interface cable, which attaches at location J3. A software library with examples is available for download; start here: http://www.evilmadscientist.com/go/PeggyArdLib

As a second option, a 6-pin ISP interface (J1) lets you program the board using an in-system programmer, for example the USBtinyISP by Adafruit Industries. This interface is supported through the AVR-GCC toolchain, and can also be used to program though the Arduino IDE with minor modifications.

Essential tools: Needed to build the kit:

1. Soldering iron + solder

A basic soldering iron meant for electronics, with a reasonably fine point tip. We recommend one of this design-- a "pencil shape" soldering iron (not gun!) with a base that holds the iron and a wet sponge. A tip in good condition (a "tinned" tip) should get shiny when hot-- able to melt and wet to solder.

While you don't need an *expensive* one, the iron *can* make a big difference in the time needed to build the kit. (Seriously. If you use one that is old and busted, or a $10 radio shack iron, or that thing from the dollar store, please expect to spend at least twice as long soldering!)

Our recommendation for a low-cost iron:
model WLC100 by Weller, about $40.

You'll also need some solder. Thin *rosin-core* solder (roughly .020 - .040" in diameter) is the most common type for electronic soldering, and is the only choice that is appropriate for electronic kits. Either standard (lead-bearing) or newer "lead free" solder types will both work just fine.

2. Angle flush cutters

For clipping loose wire ends close to the circuit board.

e.g., Sears Craftsman

3. Alkaline D-Cell batteries (3)

Besides batteries, Peggy 2 can also run from an external *regulated* power supply, 4.5 - 5 V DC, rated for at least 700 mA.

Using an inappropriate voltage or polarity can cause permanent damage; please be careful if you choose to use your own power supply.
(Yes, you can get replacement parts, but it's a hassle. Dig?)

4. Small flathead screwdriver

For adjusting brightness, when you're all done.

Optional but suggested:

1. Resistor lead forming tool

Allows fast, neat bending of resistor leads.

This one is Speedy Bend 801, Mouser part #5166-801 (~$8). (And also one of the most popular items at the Evil Mad Science shop!)

2. DIP IC lead forming tool

Bend those IC leads straight to put them in the sockets. Not really a big deal, but....

e.g., Jameco 99363: ~$8

And for Programming...

Having a programming cable for Peggy 2 is strongly recommended.

1. USB-TTL Cable

FTDI model TTL-232R or equivalent. A "smart" converter cable with a USB interface chip inside. One end hooks up to your USB port, the other to Peggy 2. This allows you to program Peggy 2 through the Arduino development environment (http://arduino.cc/).

Alternately, Peggy 2 can be programmed through any AVR ISP programmer, such as the USBtinyISP.

2. Computer, Internet access, USB port....

All of the software that you'll need is available online for free. You'll need a reasonably recent vintage computer (Mac, Windows, or Linux) and internet access.

Get started here: http://www.evilmadscientist.com/go/peggy2

This table lists the parts that go together to make a Peggy 2, roughly in the order of assembly. It's a handy reference. However, it is *not* (repeat: ***not***) a set of build instructions! There are a few places where the operations and their order is important for (possibly) subtle reasons. Please follow along as we go through the steps, even if you are an expert.

(Most of these parts are included with the kit. This table also lists the LEDs-- *you get to choose your own*-- plus the optional button set and power supply varieties.)

The kit contents are organized for clarity.

Most parts are either unique looking or otherwise easy to identify. For example, while there are two similar types of small capacitors (#'s 9 & 17), the two types are separately bagged, and each set is taped together.

Parts are labeled by their line item number from this table, and will be referred to that way in these instructions. Part **#1**, for example is the circuit board itself.

Not every location on the circuit board will be filled with a component-- just the ones that are on the list here, so don't panic!

Line	Designation	Value	Type	Digi-Key#	QTY
1	PCB	Peggy 2.3		N/A	1
2	RA4	10 k	Resistor, 1/6 W	10KEBK-ND	1
3	RA1, RA2	1k	Resistor, 1/4 W	1.0KQBK-ND	2
4	RB0-RB24	620 Ohm	Resistor, 1/4 W	620QBK-ND	25
5	U1 (Socket)	28-pin DIP socket		3M5480-ND	1
6	U2,U3,U4,U5 (Sockets)	24-pin DIP socket		3M5478-ND	4
7	VR1,VR2	5k trimpot		490-2888-ND	2
8	XTL	16 MHz	oscillator crystal	631-1108-ND	1
9	C1,C2	18 pF	Capacitor, ceramic	BC1004CT-ND	2
10	C12	100 uF, 10 V	Cap., electrolytic	P5123-ND	1
11	S1, S2	Tactile Button Switch	B3F-1000	SW400-ND	2
12	L2, Battery box leads, plus ((JP1 & JP2) OR (JP3 & JP4))	Zero-ohm jumpers (Look like resistors with one black stripe.)		0.0QBK-ND	5
13	J1	6-pin DIL header	ISP connector	609-3210-ND	1
14	J3	6-pin SIL header	TTL Connector	609-3291-ND	1
15	Q0-Q24	2STX2220	Transistor	497-7067-ND	25
16	C3	4700 uF, 10 V	Cap., electrolytic	P5130-ND	1
17	C4-C10	0.1 uF	Capacitor, ceramic	BC1148CT-ND	7
18	S3	high-power slider, SPDT		CKC5107-ND	1
19	U1	ATmega328P Microcontroller (pre-programmed in kits)		ATMEGA328P-PU-ND	1
20	U2,U3	CD74HC154EN demultiplexer		296-9181-5-ND	2
21	U4, U5	STP16DP05B1R LED driver, or exact substitute		497-5974-5-ND	2
22	Battery Box	3 x D cell		BH3DL-ND	1
23	Cable Ties	Cable ties, 4.5"x0.1", black		RP202C-ND	2
-	DXXYY	Through-hole LEDs-- *not included as part of the Peggy 2 base kit.*		N/A	*up to* 625
24	Rubber feet	McMaster Carr 95495K66		N/A	6
-	All others	Leave empty or hack things in, at your discretion.			

	Optional extra button set			
1	b1,b2,b3,b4,b5	Tactile Button Switches (optional)	SW400-ND	up to 5

	Optional Power adapter kit: US plug		
1	Power adapter: US	5 V DC regulated, 1 A, 2.5 mm plug	T977-P6P-ND
2	J2	2.5 mm power jack, US power jack	CP-002B-ND

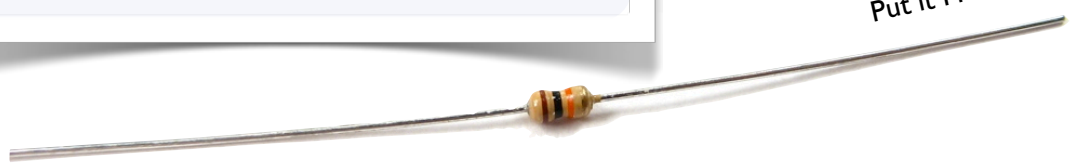
	Optional Power adapter kit: International multiplug		
1	Power adapter: Int'l	5 V DC regulated, 1.2 A, 2.1 mm plug	T946-P5P-ND
2	J2	2.1 mm power jack, Int'l power jack	CP-002AH-ND

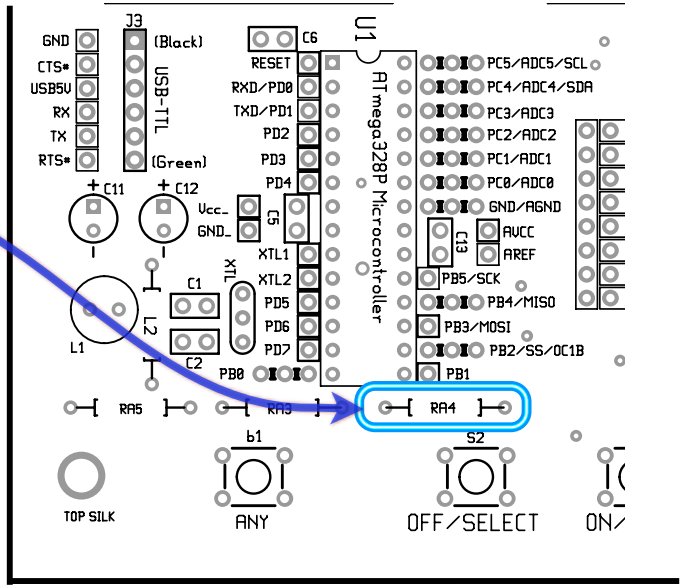
Part **#2** is a 10 kilo-ohm resistor

(Color code: Brown-Black-Orange-Gold)

Also: it's the one smaller-size resistor!

Add one resistor to the circuit board, in location **RA4**.





(Lower left corner of printed circuit board)

Implied procedure for adding electronic components:

1. Bend the leads of components as needed.
   (Resistors need to be bent, most others parts do not.
   Optionally use lead forming tool shown in Step 1.)

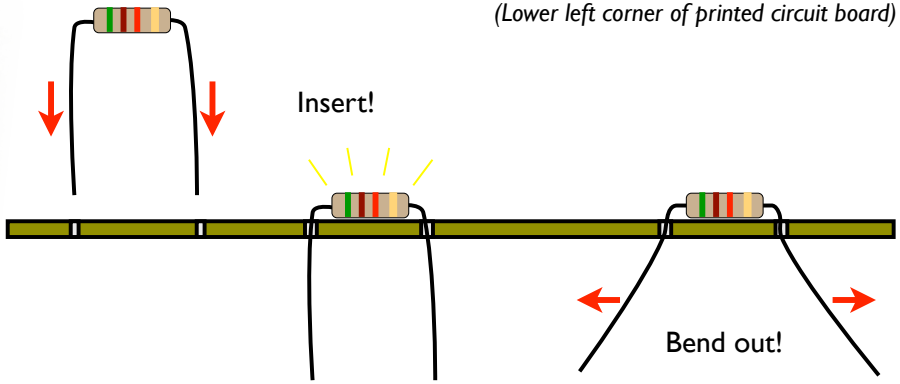
2. Insert each component into the circuit board, from the top, at its given location. Push it flush to the board.
   (Resistors are *unpolarized*; they can go in *either way*.)

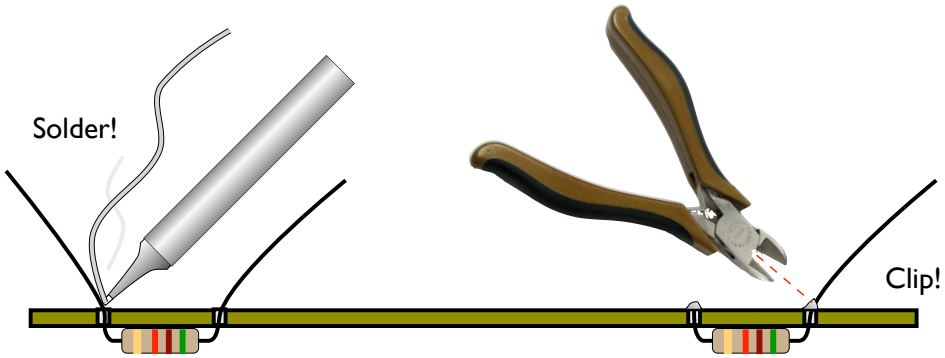
3. On the back side, *gently* bend the leads out at 45° to hold components in place while you solder.

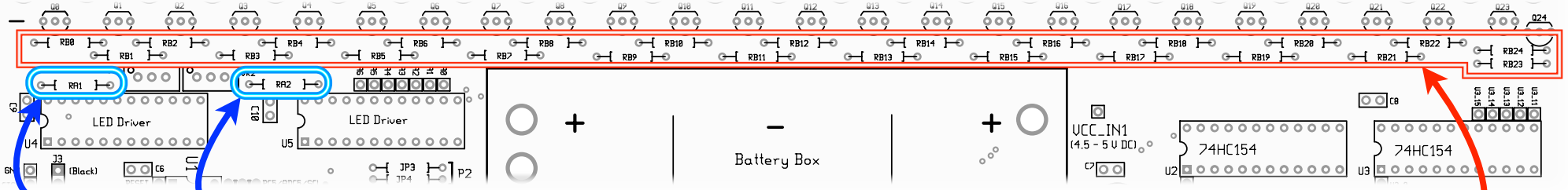
4. One at a time, from the back side, solder the leads of the component to the circuit board.

   - Your tip should be shiny (tinned). If not, melt some fresh solder against it and wipe clean on a wet sponge.
   - Place the solder against the joint that you wish to connect.
   - Touch the iron to the solder and joint for about one second. Count it out: "one thousand one."
   - The solder should melt to the joint and leave a shiny wet-looking joint. If not, let it cool and try again.

5. Clip off extra leads on back side, flush to the board.
   (But not so flush that you're clipping the board itself.)







Part **#3** is a 1 k ohm resistor

(Color code: Brown-Black-Red-Gold)

Install two resistors, in locations **RA1** and **RA2**.

Part **#4** is a 620 ohm resistor

(Color code: Blue-Red-Brown-Gold)

Add 25 of these to the circuit board, in locations **RB0** through **RB24**.

Part **#5** is a 28-pin DIP socket.

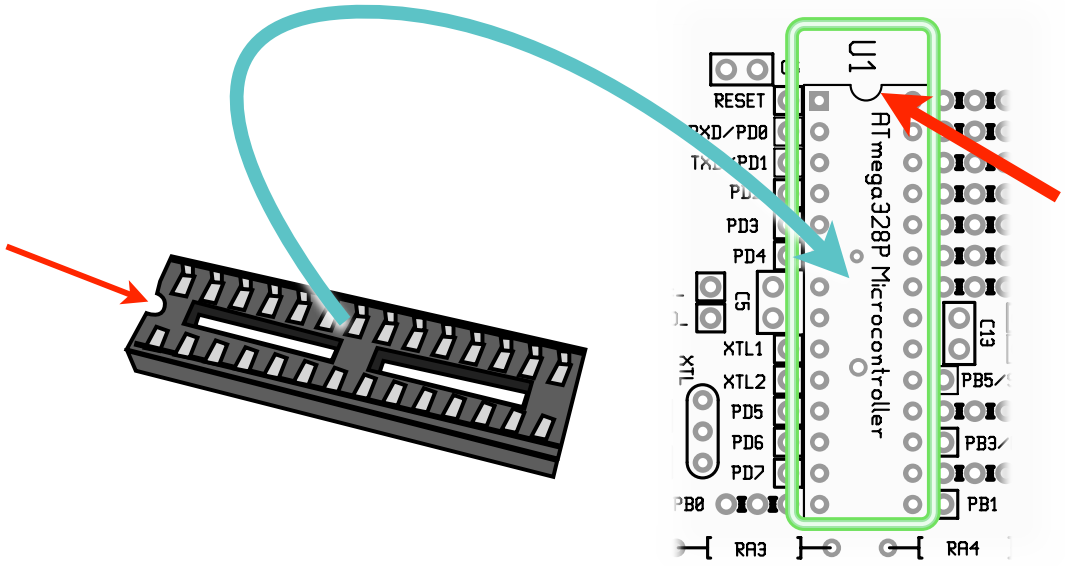
(That's the *long* one.)

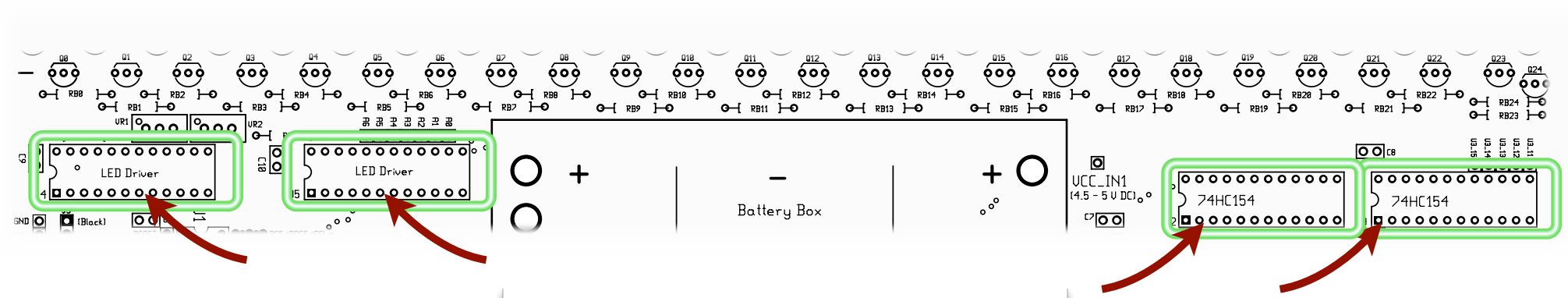
Install this socket in location **U1**.

Orientation matters: Match the "half-moon" shape at one end of the socket to the one drawn on the circuit board.

Seat the socket flush onto the board and bend back a couple of pins on the bottom side to help hold it in place while you solder. (Alternate method: Hold it down with a piece of tape while you solder the first couple of pins.)

Solder every pin of the socket in place.
We'll install the chip in this socket later.





Part **#6** is a 24-pin DIP socket.

Install these sockets in location **U2**, **U3**, **U4**, and **U5**.

Follow the same procedure as for the last socket; match the half-moon end of each to the drawing.

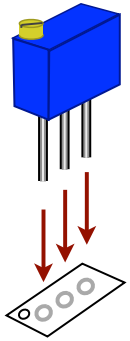
Part **#7** is a 5 k trimpot

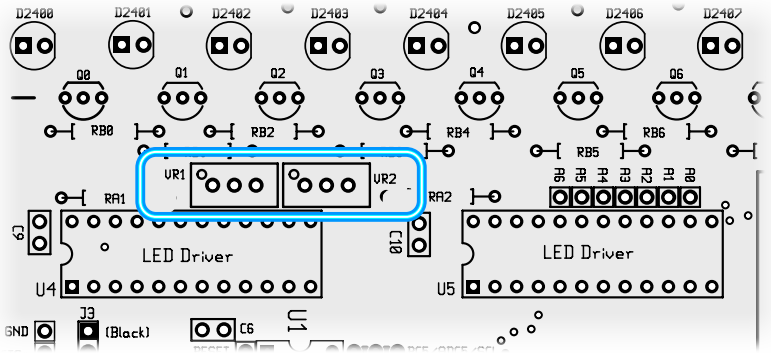
(A little blue box with three pins)

Install two of these in locations **VR1** and **VR2**.

Orient the corner with the brass screw over the corner of the drawing that has a little circle.

Seat it flush to the board, and bend out the pins, *gently*, to hold it in place while you solder them.

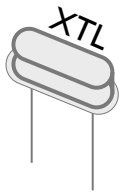


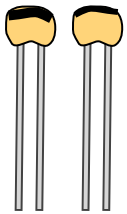


Part **#8** is a 16 MHz crystal oscillator

(A shiny metal capsule with fangs.)

There is only one of these, which goes in location **XTL**. The two pins go in the outer two holes; ignore the middle hole. (*Orientation*: Either way.)





Part **#9** is an 18 pF ceramic capacitor.

(A little yellow bead with two pins & a black mark.)

Add two of these capacitors in locations **C1** and **C2**. (Like resistors, you can put these in either way.)

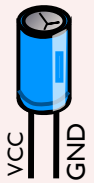
The labels on these caps are *gosh-darned tiny*. They're the ones bagged together and marked with a black stripe.

Part **#10** is a 100 µF electrolytic capacitor

(A small plastic-covered cylinder; color may vary.)

There is only one, and it goes in location **C12**.

Orientation matters. The NEGATIVE side is marked with a broad white stripe. That side goes to GND, the other side to VCC.



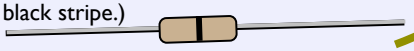
Part **#11** is a tactile button switch

Two of these *snap* into locations **S1** and **S2**.

Orientation: Match the shape of the switches to the drawing on the circuit board; the pins are on the left and right sides of the buttons, not top and bottom. Solder all four pins of each switch.

Part **#12** is a jumper

(A resistor-looking thing with one black stripe.)



Bend it like a resistor and install it in location **L2**.

(*Orientation*: Either way.)

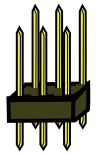
Parts **#13** & **14** are 6-pin headers.

These connectors are only needed if you plan to reprogram your Peggy with an external interface.

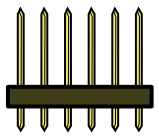
J1 is a 6-pin dual-inline (DIL) header for use with an AVR-ISP programmer. **J3** is a 6-pin single-inline (SIL) header for use with a USB-TTL cable

Orientation: the ends with short pins go down into the board.

These headers fit loosely in the holes; they will not stay in place when you turn it upside down. Do not try and bend the pins to hold it in place (they are short and stiff!) but instead rest the board on the header to solder each one.



6-pin DIL ("2x3") header (J1)

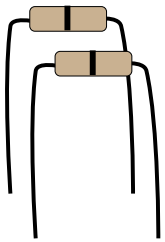


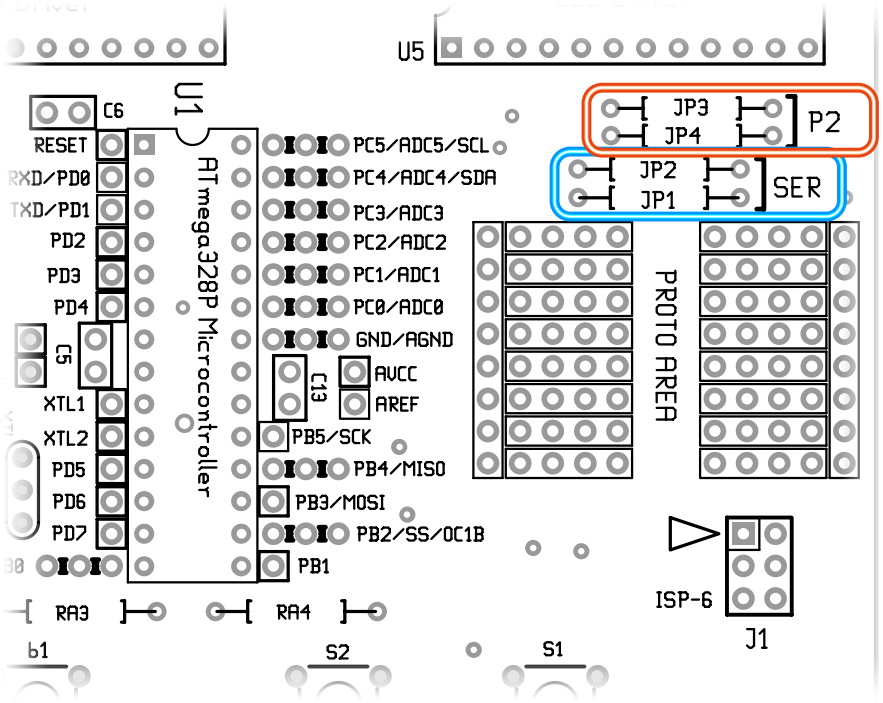
6-pin SIL ("1x6") header (J3)

Next, two more jumpers...

Two more wire jumpers (part **#13**) are required on the board for configuration.

Normally, the jumpers go in locations JP3 and JP4, which builds the Peggy for full compatibility with existing Peggy 2 code. The initial firmware that comes on the microcontroller assumes that the board is built this way.





As an advanced option...

If you put the jumpers in locations JP1 and JP2, this enables the serial port, but *breaks compatibility* with most existing Peggy2 code.

On the other hand, software examples for this modification *are* available, and *most* existing Peggy2 code can be made to run by using a different version ("Peggy2serial") of the Peggy2 Arduino library.

There are additional changes to note if you use this modification. Pins RXD/PD0 and TXT/PD1 are normally used as part of the row-selection process for multiplexing the display. To free those lines up, lines PC5/ADC5/SCL and PC4/ADC4/SDA are used instead. The PORTC lines are normally used for I2C (TWI), ADC, or GPIO (usually hooked up to read out button locations b1-b5). There's also some additional system overhead, but generally not enough to be concerned about.

To summarize the main tradeoffs:

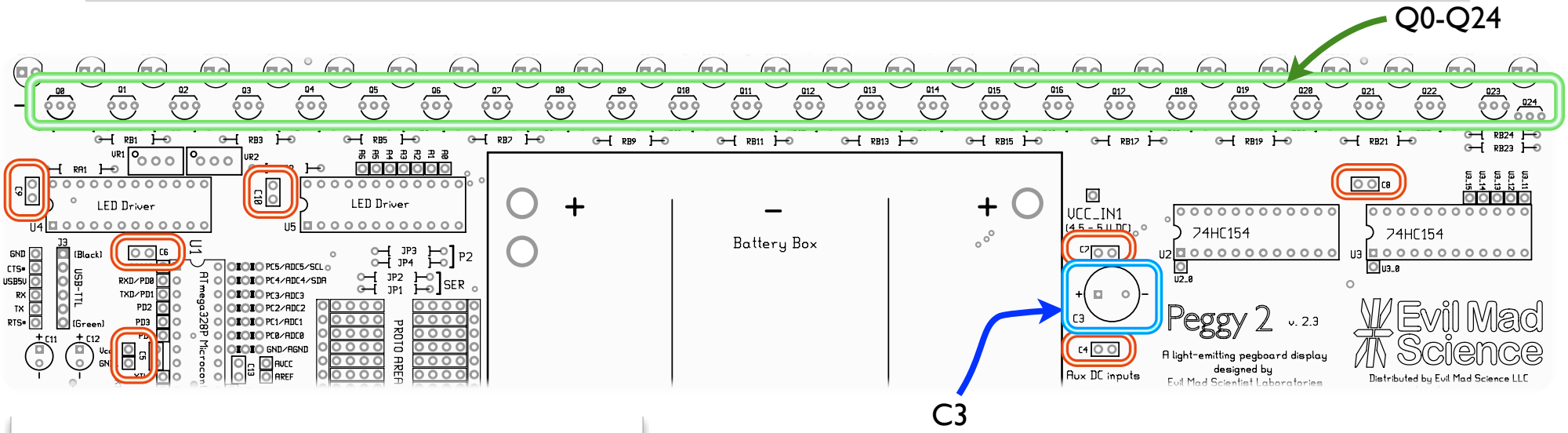
"**P2**" -- Normal, full Peggy2 compatibility.
- Ready to test full hardware without reprogramming
- All buttons b1-b5 available*
- I2C/TWI port available*
- Up to 6 ADC channels available*
- Serial port available for programming only (*not* during display time)

*These resources share the same pins.

"**SER**" -- Serial port modification
- Needs reprogramming to test out all hardware
- Buttons b1-b4 available*, b5 *not* available
- I2C/TWI port *not* available
- ADC4/ADC5 *not* available, up to 4 available*
- Serial port always available
- Slightly lower data rate

* These resources share the same pins.

(Again, this an advanced option. We suggest it only for advanced users.)

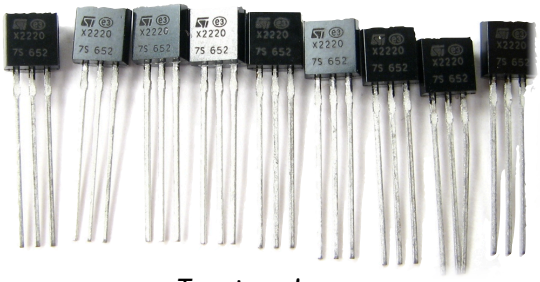


Part **#15** is a type 2STX2220 transistor

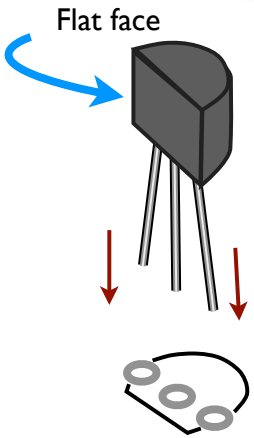
Install 25 of these little tripods in locations **Q0** through **Q24**.

Orientation matters. Match the flat face of the transistors to the drawing on the board.

Bend the pins out slightly to fit in the three holes; they will not go all the way flush to the board, but will sit somewhat above it. (*Be gentle*, do not force it!) You can bend the pins out slightly, on the bottom side, to hold it in place while you solder.



Transistors!

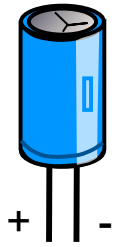


Part **#16** is a 4700 µF electrolytic capacitor

(A BIG plastic-covered cylinder.)

There is only one of these; it goes in location **C3**.

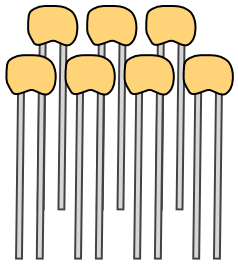
Orientation matters. The NEGATIVE side of the capacitor is marked with a broad white stripe. Solder it with this negative side towards the "-" on the circuit board.

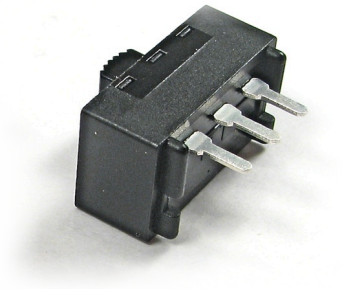


Part **#17** is a 0.1 µF ceramic capacitor

Add seven of these capacitors, in locations **C4**, through **C10**. (Orientation: Either way.)

Again, the labels are tiny. If your eyes are very good, you *might* be able to make out the legend "104" on these. But no need. :)

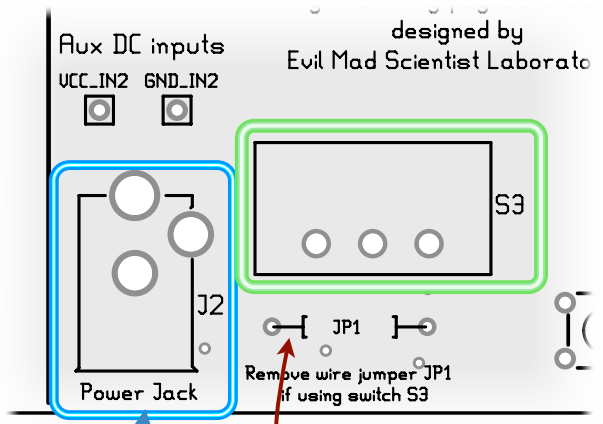




Part **#18** is a high-power slider switch

This switch goes in location **S3**, oriented so that it matches the drawing.

It fits only loosely into its holes, and you should not bend its pins. You will need to rest the board on it, tape it, or otherwise hold it while you solder it in place.



JP1 is an unused location to "hotwire" across the power switch. (It should be left *empty*.)

Location for optional external power jack

Note: If you are adding external DC power on your own, make sure make sure it's 4.5 - 5 V DC, center positive, *well regulated,* protected, and capable of sourcing *at least* 700 mA. *If you aren't sure, don't plug it in.*

Part **#19** is an ATmega328P microcontroller

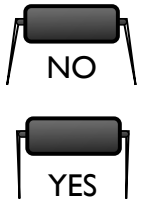
This chip sits in the 28-pin socket (part **#5**), in location **U1**.

Orientation-- *very important* --Half-moon end of chip matches that of socket and the drawing on the circuit board

If necessary, bend the leads of the chip to vertical-- straight up and down-- before inserting it into the socket. Do not bend them by hand; bend all pins on one side at a time by pushing them against a hard flat surface.

The chip goes into the socket with firm, even pressure.

From end of chip:



Part **#20** is a CD74HC154 demultiplexer

These two chips sit in the 24-pin sockets at locations **U2** & **U3**.

Follow the guidelines given for part #18; match the orientation, and straighten the pins if necessary.

Part **#21** is a STP16DP05 LED driver chip

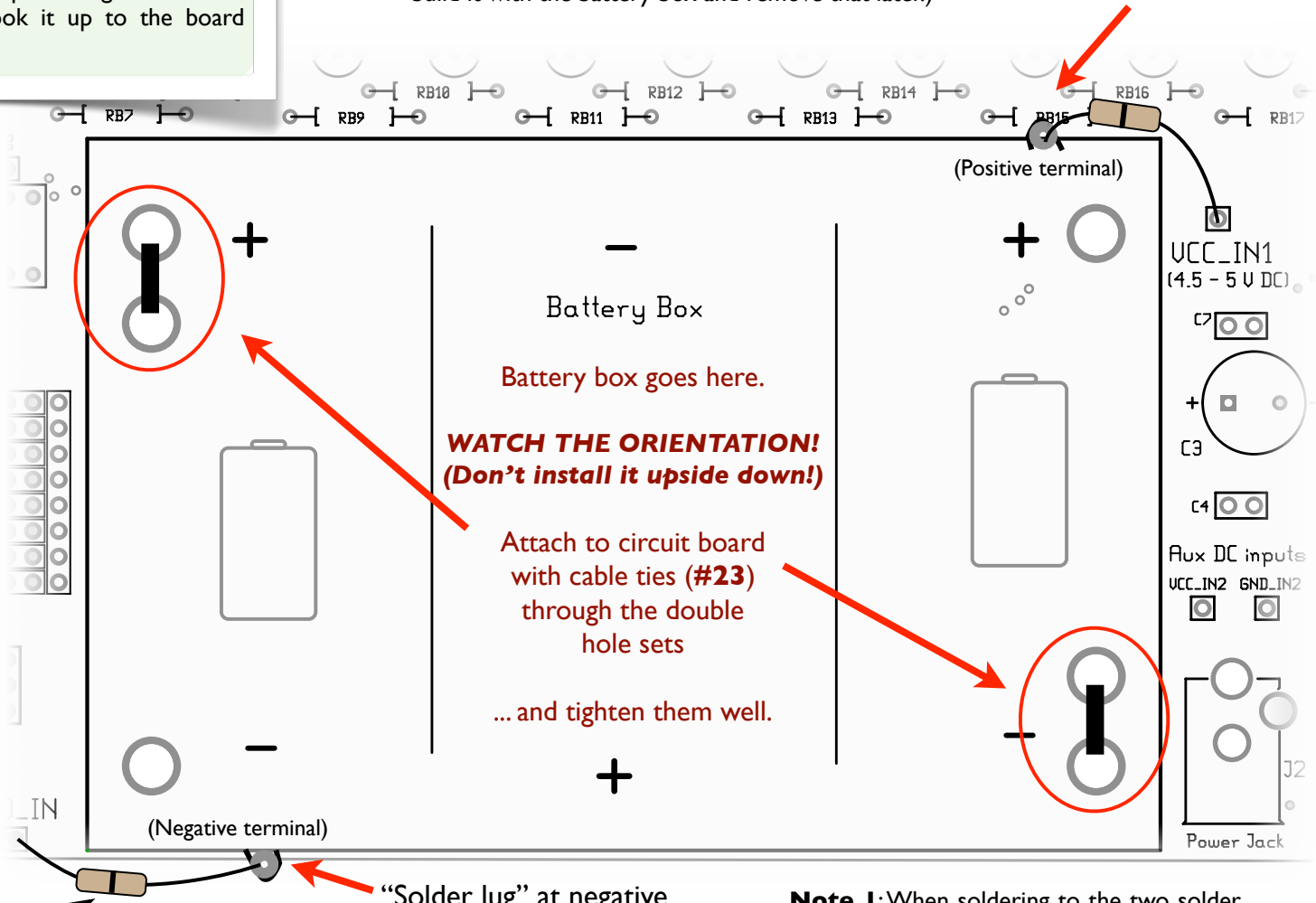
These two chips sit in the 24-pin sockets at locations **U4** & **U5**.

(You should know the routine by now!)

Part **#22** is the battery holder

It holds three D' Cells and lives in the spot marked "Battery Box." First attach it in place-- tight --with the cable ties (**#23**), and then hook it up to the board through a pair of wire jumpers.

(*Note*: If you have an alternate power source, e.g., you plan to run from an AC adapter all the time, you *can* skip this step without ill effect. Or build it with the battery box and remove that later.)



Note 1: When soldering to the two solder lugs, be quick-- the plastic of the battery box can (and will) melt if you aren't careful.

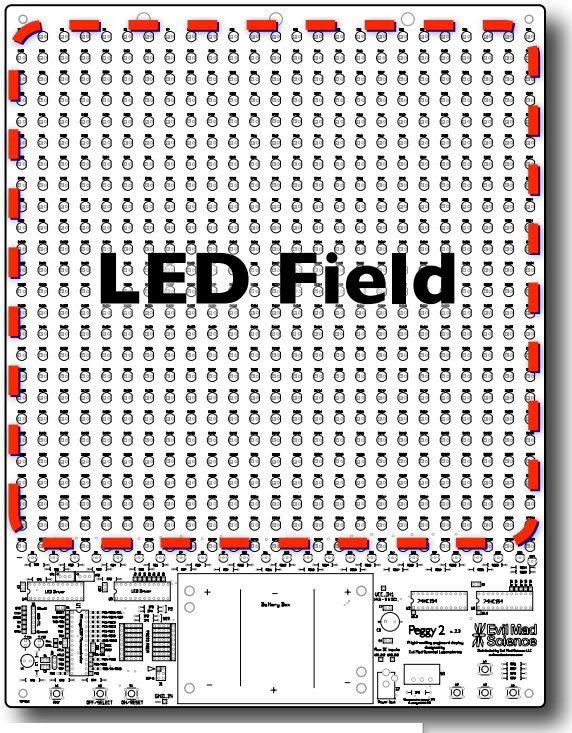
Note 2: The *number one* cause of people needing to purchase replacement Peggy chip sets: Installing the battery box upside down. *Please* make sure it's right side up!

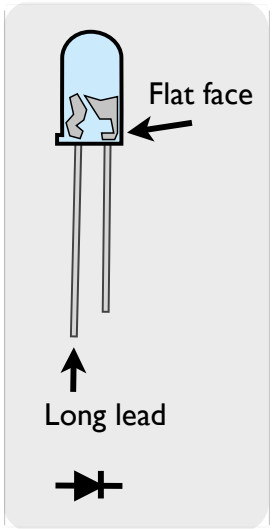
Next, it's time to add the LEDs.

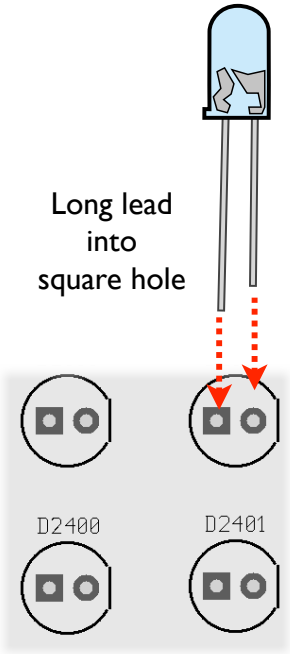
Peggy 2 accommodates *up to* 625 LEDs in standard sizes up to 10 mm. 3 mm, 5 mm, and 8 mm LEDs will work just fine. Put them where you like, or everywhere.

For standard types of LEDs, the long lead goes in the square hole (the one on the left), and the flat face of the LED package (if any) matches the drawing on the circuit board.

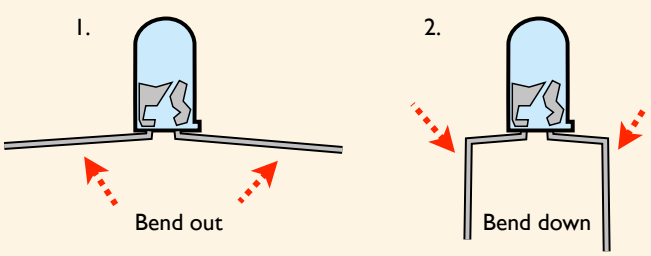
For reference, each LED location is labeled DXXYY, where XX is the row number and YY is the column number. If you do not fill all the holes and have an uncontrollable urge to blacken the unused labels, a black permanent marker works well.





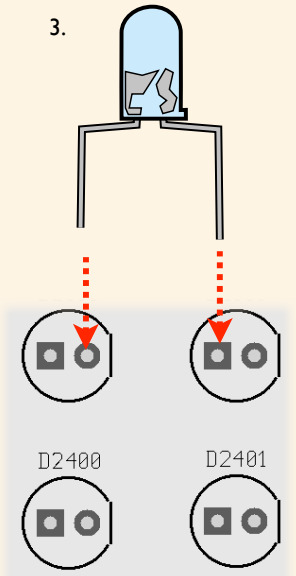


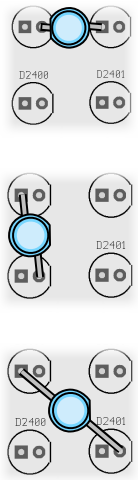
Alternative build idea: put all the LEDs on the back side of the circuit board for truly bare background. In this scheme, the long lead *still* goes in the square hole.



Where the grid *really* won't do, you can put LEDs *between* grid locations. Which ever way you do it, the side of each LED with long lead *still* goes to square hole. Side with flat *still* goes to round hole.

>>This procedure is recommended mainly for static signs and LED displays; putting LEDs between grid locations can make programming those grid locations less intuitive.





Last component: Add the rubber feet, part **#24**

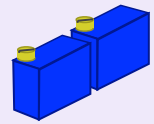
Attach one rubber foot in each corner on the back side of the circuit board, and a couple closer to the middle, to protect against bending when you press buttons-- make sure that the circuit board lies flat on these bumpers, not on wire leads.

The feet will help to avoid accidental short circuits, as well as protect your wall if you hang it up by a hook or string.

Last required step: Brightness Adjustment

Trimpots VR1 and VR2 can be used to adjust the brightness of the left and right "halves" of the display. VR1 controls columns 0 through 15, and VR2 controls columns 16 through 24. Usually, you want to approximately match the brightness of the two sides.

The adjustment knob on each is a tiny brass screw with 20 turns of range. It lets you set Peggy somewhere between burn-the-batteries bright and power saving dim.



Need help?

If you encounter difficulty with Peggy 2 in hardware, software, or elsewhere, odds are that somebody knows how to help you out. Your first stop should be the Evil Mad Scientist Laboratories forums:

http://www.evilmadscientist.com/forum/

Big troubleshooting hint:

90% of assembly issues are caused by one of the following three things:

1. Component missing or in the wrong location.
2. Backwards component
3. Bad or missing solder joint.

An open-source project

The hardware and software designs used in this project are being released under an open-source license. For more information, please see:

http://www.evilmadscientist.com/go/peggy2

Example firmware is available for download, and we'd love to see what you can do with it!

Got pictures?

If you have interesting pictures or video of things built using this kit or the hardware or software designs, we'd love to see them in the Evil Mad Science Auxiliary:

http://www.flickr.com/groups/evilmadscience/