DIY LED Cube 4x4x4

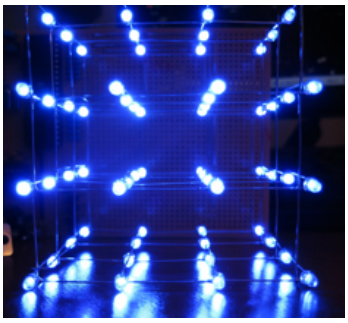
Amazing 3D display with endless possibilities

By Ari Dubinsky and Brad Eckert
Product Marketing Assistants

Description: Build an amazing 3D LED display
Difficulty: Intermediate-Advanced
Time Required: 6 hours

Knowledge Required: Basic electronics and soldering skills, AVR microcontroller programming skills.

64 LEDs makes up this 4x4x4 cube, controlled by an Atmel Atmega16 microcontroller. Each LED can be addressed individually in the software, enabling it to display amazing 3D animations!



Ari's Amazing 3D LED Display

LED Cube Kit (P/N 2146329)
Kit includes:

(1) Prototyping Board PN 206594
(1) Atmel AVR Atmega 16 Microcontroller- 323133
(64) Blue Diffused 3mm LEDs PN 333383
(18) 220 Ohm Resistors PN 690700
(1) Red LED PN 333973
(1) Green LED PN 34761
(1) 10k Ohm Resistor PN 691104
(4) 2.2k Ohm Resistor PN 690945
(4) BC337 Transistors PN 254810
(3) 0.1uF Capacitor PN 544921
(2) 22pF Capacitor PN 15405
(1) 14.7456MHz Crystal PN 324187
(2) SPST Buttons PN 155380
(1) 40 Pin IC Socket PN 112311
(1) 10 Pin Header PN 67812
(1) ISP Programmer PN 2136788
(1) USB Cable PN 222010
(1) Hookup Wire PN 2152884
(1) Coin Cell Battery PN 2123784

Optional Materials:

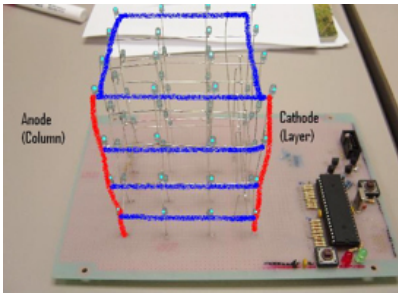
(1) 10uF Capacitor PN 29891
(1) 1000uF Capacitor PN 330722
(1) 7805T Transistor- 51262
(1) Battery Clip PN 11280
(1) 9V Battery PN 198731

Tools list:
Wire Cutters PN 35482
Pliers PN 177608
63/37 Solder PN 151474
Solder Flux PN 2094258
Solder Wick PN 41082
25W Soldering Iron PN 129040
Soldering Stand PN 36329
Sponge PN 134631
LED Tester PN 355805

Used to build optional circuit that will supply 5V of power with an 9V source.

LED cubes rely on an optical phenomenon called persistence of vision or POV. If you flash an LED really fast, the image stays on your retina for a little while after the LED turns off. By flashing each layer of the cube one after another really fast, it gives the illusion of a 3D image, this is also called multiplexing.

The LED cube is made up of columns and layers. Each of the 16 (anode) columns and the 4 (cathode) layers are connected to the controller board with a separate wire, and can be controlled individually.



Anode (red) & cathode (blue) layers

Step 1 - Pick a good time
You will need a lot of time to solder together 64 LEDs.

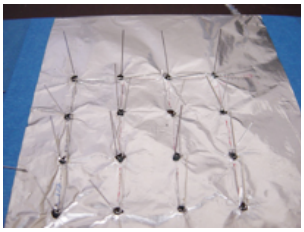
Step 2 - Build your LED layers
Soldering grids of 4x4 LEDs freehand would look terrible. To get even-looking 4x4 LED grids, we'll use a template to hold them in place. Also, to minimize adding or cutting wire, we'll use the LED legs to connect the LEDs together.

A1. Cube template (the easy way): -Find a piece of pegboard that already has a 4x4 grid pre-drilled with 1 inch spacing between holes -Double fold a piece of aluminum foil over the board and tape it down. The foil will hold the LEDs in place and protect the board while soldering -Use one LED to punch an LED-sized hole through the foil for each hole

A2. Cube template (the not as easy way): -Find a piece of wood large enough to make a 1inch 4x4 grid (leave a little extra room) -Draw up a 4x4 grid of lines with a spacing of 1 inch. -Make dents at the intersect points with a center punch -Drill 16 holes small enough so that the LED will stay firmly in place, and big enough so that the LED can easily be pulled out (without bending the

wires)

B. Place your LEDs in the template to be sure they are properly spaced.



Pegboard/foil template

Wood template

Step 3 - Test each LED
For obvious reasons, it is vital to have functioning LEDs. I found out the hard way, it's much easier to test the individual LEDs before you solder them together. Sticking an iron to desolder a damaged LED in the middle of your cube is as tough as it sounds. Take the time to test them.

You can hook the LEDs up to a 3 volt power supply and briefly powering on, use an LED Tester, or simply use a coin cell battery. Hold the coin cell between the legs of the LED and then squeeze the legs. You don't need a resistor since the coin cell runs at 3 volts and you are only touching it for couple seconds.

Step 4 - Soldering LED layers

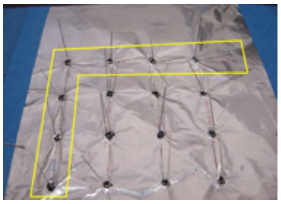
A. To make the cube's four layers of 4x4 LEDs, bend their cathodes (the shorter lead) and solder them together. You get to learn from my mistakes. Here are some soldering tips I have learned to follow.

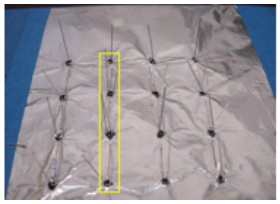
Soldering iron hygiene. Keep the soldering iron clean. That means wiping it on the sponge every time you use it or whenever you see the tip becoming dirty with flux or oxidization, even if you are in the middle of soldering. Having a clean soldering tip makes it A LOT easier to transfer heat to the soldering target.

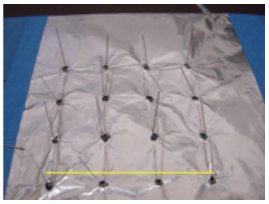
Soldering speed. Get in and out quickly. Apply a tiny amount of solder to the iron tip. Touch the part you want to solder with the side of your iron where you just put a little solder. Let the target heat up for 0.5-1 seconds, and then touch the other side of the target you are soldering with the solder. Remove the soldering iron immediately after applying the solder.

Mistakes and cool down. If you make a mistake, for example if the wires move before the solder hardens or you don't apply enough solder. Do not try again right away. The LED is already very hot, and applying more heat with the soldering iron will only make it hotter. Continue with the next LED and let it cool down for a minute, or blow on it to remove some heat.

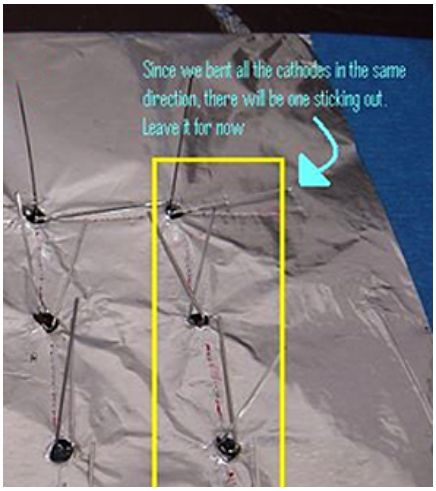
B. Create your layer:
• Place LEDs in template on two outer rows in an "L" shape (see picture below) and solder them together
• Continue to insert LEDs row by row and soldering them together (going one row at a time leaves you space to solder) until you complete the rows
• Add wire cross bracing in the front where the led rows are not connected (use the same hookup wire but strip the plastic coating off and straighten the wire).

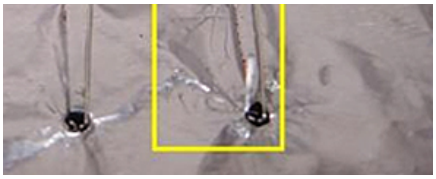




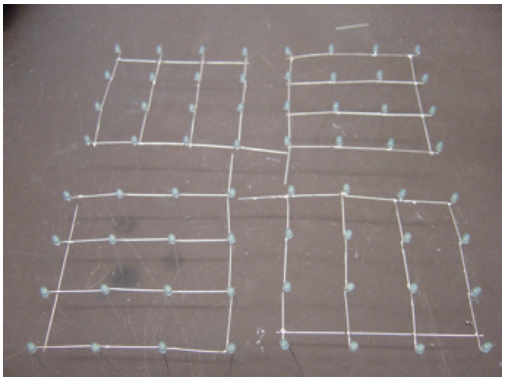


• Straighten the upright LED legs
• Don't remove the tab from upper right LED - you'll use this later





Leave the corner leg – you'll use it later



Assembled layers

Step 5 - Solder the layers together

Take your time while building the layers. The quality and look of your final cube depends on the layers be built neatly and evenly.

A. Select your best layer piece and put it back in the template. This will be the top layer.

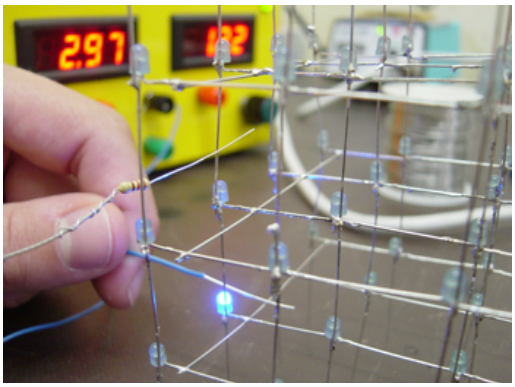
It can get tricky holding one layer above the other while soldering. You can use a third hand, or I used a 9V battery which is the perfect size to help create the correct spacing.

Warning: Tape over the battery poles to avoid accidentally overloading the LEDs while soldering.

B. Carefully align the layers and solder the corner LEDs. Next, solder all the LEDs around the edge of the cube, moving the 9V batteries along as you go around ensuring that the layers are soldered in parallel. Then move a 9V battery to the middle of the cube, sliding it in from one of the sides and solder a couple of the LEDs in the middle.

C. Your cube should be fairly stable now, so you can continue soldering the rest of the LEDs without using any extra support. When you have soldered all the columns, it is time to test the LEDs again. Remember that tab sticking out from the upper right corner of the layer? Now it's time to use it to test your LEDS.

I used my own bench top power supply but feel free to use other 3 volt sources for testing. I recommend fixing two wires on a 3 volt battery with tape and using that to test. Touch the negative wire to the layer you want to test and touch the positive wire to the column you want to test. You should see an individual LED light up. Continue touching each column in each layer to be sure they are all functioning.

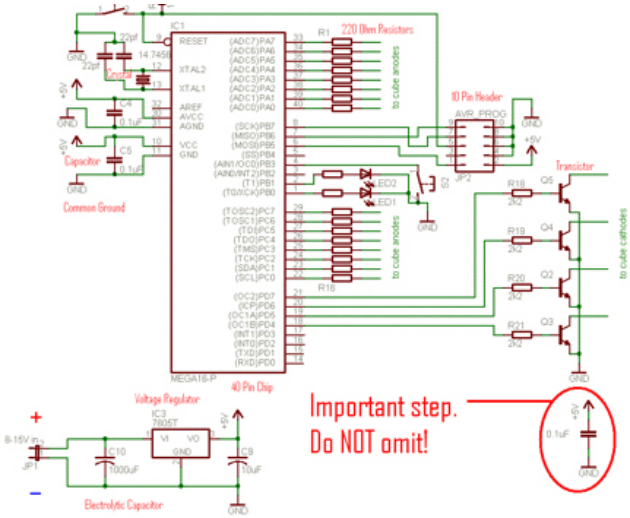


Testing the LEDS

D. Solder and test the remaining layers. Congratulations, you're half way there!

Step 6 - Building your cube's circuit

The circuit controlling the LED cube is shown in the schematic image below. I modified the original schematic to reduce clutter and confusion. Take the time to read and fully understand this schematic. I found the construction of the controller to be the most time consuming and trickiest part of the project.



Click image to enlarge

I found it helpful to break this schematic into three parts: Connection, Power, and Miscellaneous.

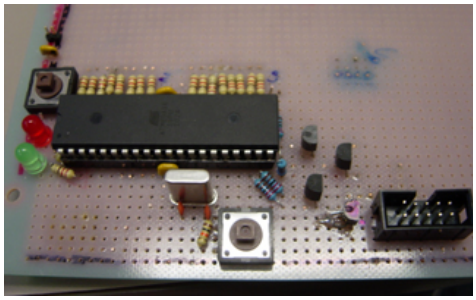
Connection: This is how we "link" the brains to the cube. Pins 22-29 and 33-40 are used to connect the 16 anode columns to the ATmega16. Pins 18-21 connect to transistors controlling the power to each of the 4 cathode layers. It's important to consider which pins go to certain columns and layers, but we will revisit their physical connection in the next step.

Power: Power should be supplied to the cube by means of "busing". With busing we can connect several terminals to each other through common strips that run along the side of the board. These bus strips come in handy when wanting to supply power because they localize all the power and ground connections to a single point so that you can reduce the number of wires running underneath your board. Mark off bus strips with a colored marker to help identify them.

To power your cube, you can either attach the AVR programmer (jumper ON) to the 10-pin header, or use an external power source. Using the programmer is really easy; attach the cube to the programmer and connect the programmer to a computer. If your soldering is good and you have the correct drivers installed this little programmer should fully power your cube. Using an external power source, (9V battery), your cube is mobile, which requires building an annex to your circuit (found on the bottom left corner of the schematic).

A. Now, let's build your circuit. Lay out all the components on the circuit board to minimize the amount of wires that run underneath the board. Be sure to orient the transistors, status LEDs, and polarized capacitors correctly. Identify pin 1 on the chip and header, see some orientation tips here. This video shows a trick for soldering when wires become messy.

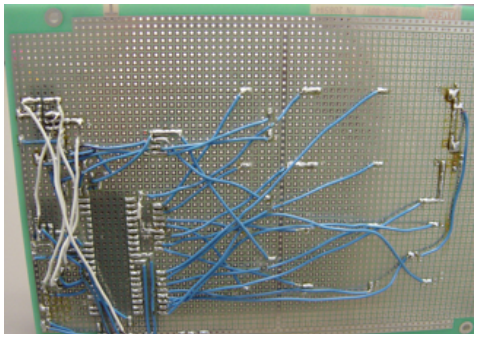
B. To reduce your chances of error, use different colored wire and keep your layout compact, on one side of the board, and neat. Use flux when soldering.



Layout the board compactly and neatly

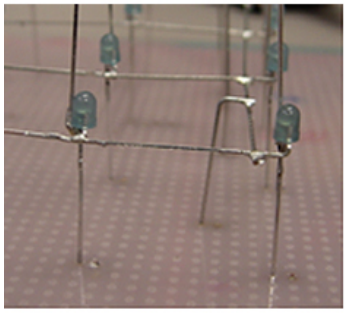
Step 7 - Wiring the board

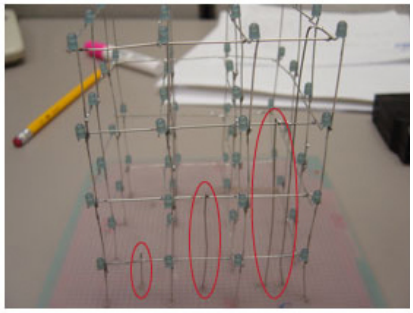
A. Time to wire up the cube. Carefully fit the cube onto the board and solder the corners, followed by the edges and then center.



A neat board layout minimizes messy wiring

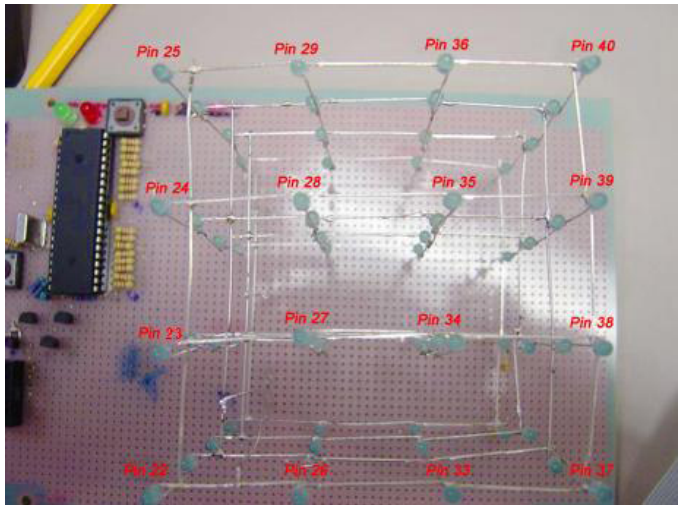
B. Next, we'll make a ground connection for each layer. This can be done by making wire hook connections to each layer as shown in the pictures below (use the same hookup wire but strip the plastic coating off and straighten the wire).





Ground connections: close-up (L) and all together (R)

C. After the cathodes have been connected to ground the anodes need to be connected to the ATmega16. The picture below shows how to connect the 16 resistors and 16 anodes in the right order.



Pin connection diagram for LED cube

Step 8 - Code and programming

You now have an LED cube. To make use of its coolness, it needs a program! I have attached a driver for rendering a 3D data space on the cube, and functions to display some cool visual effects on the cube. You can use my code, write your own or build on the code to create endless effects.

A. To compile the program, open a command prompt, enter the directory with the source code and type "make" on the command line. You should now have a file named main.hex in the source directory. The next step will show you how to get that code into your cube.

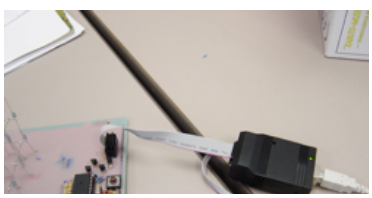
Click the icon to download files

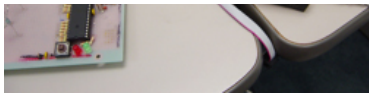
B. Time to program the microcontroller. First, download the latest version of AVRDUDE, available here.

Be careful when completing the following this steps.

If you get it wrong, you can permanently destroy your microcontroller!

1. First off, let's just see if we can make contact with the AVR. Connect the programmer to your cube and your computer.





Connect cube programmer to computer

2. Open a command prompt. Enter the command "avrdude -c usbtiny -p m16", where -c specifies the programmer, and -p the AVR model. You can see the output in the images below.

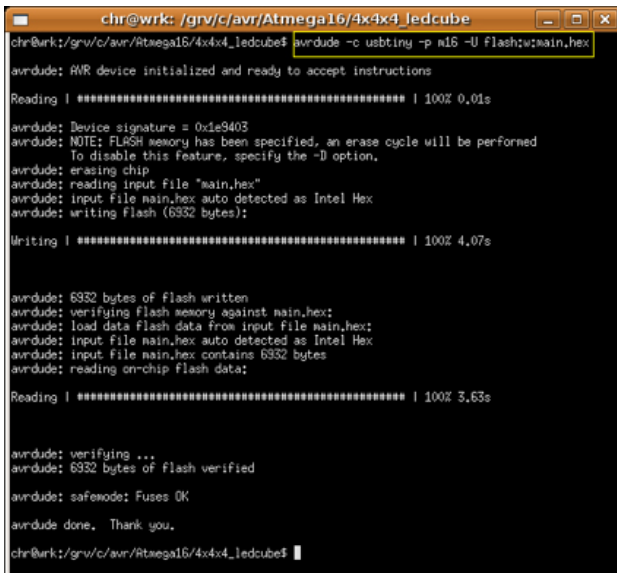
3. Now, upload the firmware: "avrdude -c usbtiny -p m16 -U flash:w:main.hex".

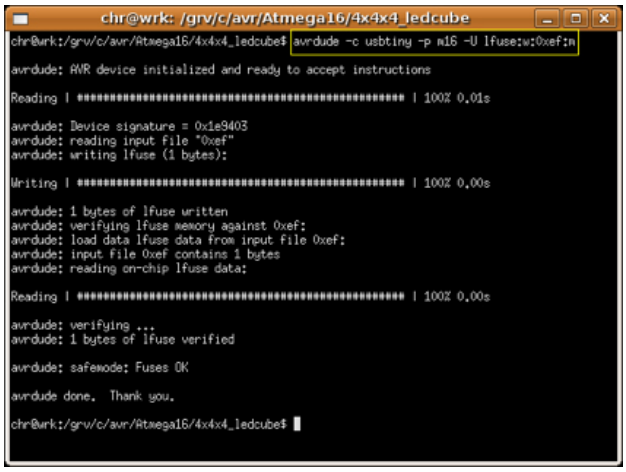
By now, the cube should reboot and start doing stuff. It will be running at 1MHz (very slowly) using its internal oscillator. Some of the LEDs won't work because some GPIO ports are used for JTAG by default.

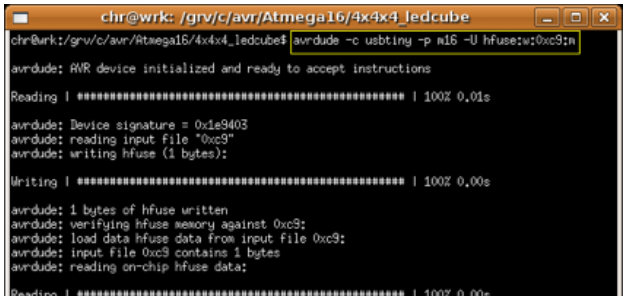
4. To enable the external oscillator and disable JTAG, we need to program the fuse bytes: Run "avrdude -c usbtiny -p m16 -U lfuse:w:0xef:m" Run "avrdude -c usbtiny -p m16 -U hfuse:w:0xc9:m".

After writing the correct fuse bytes, the cube should reboot and start operating at regular speed with all LEDs operational.

The screen shots below are what you should see if you have entered the commands properly.



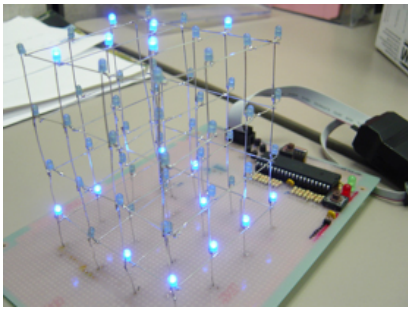




5. Enjoy your new cube!

Troubleshooting Issues:
1. Check the soldering job. Make sure solder leads don't touch each other.
2. Run a continuity test between each connection using a schematic printout
3. Run a continuity check between ground and power.
4. Make sure polarized components are oriented correctly on the board (e.g. IC, capacitors, and LEDs)
5. Are the header pins soldered correctly?
6. Check for damaged components
8. Does the green light come on when the programmer is connected?
9. The programmer should be supplying 5 volts to the cube with the jumper 'ON'.
If not, then use a different power source to test the cube.

Still need help?
Double check the original 4x4x4 LED cube recipe on Instructables here.
For lots more LED cube fun, check out Instructables contributor page "chr" here.



The options for entertainment and dazzlement are endless your LED cube.

+ You can used bigger LEDs for more brightness (be sure you use the correct value resistor)
+ Set your lights, uh, sights high, expand up to an 8x8x8 cube (you may need an additional IC or shift-registers)
+ What can you dream up?

Find the original LED cube by Chr here?

Check out a 3x3x3 LED cube build by a Jameco favorite, Joey Hudy here

---

This project was assembled by Ari Dubinsky and Brad Eckert. Ari currently attends Cal Poly San Luis Obispo in pursuit of a degree in Electrical Engineering and has teamed up with Jameco to accelerate his learning as well as to lend a helping hand to a major electronics distributor. His interests include shredding on the guitar, electronics, and music production.

Brad grew up in Bay Area and now attends MIT. He wants to become an egineer and loves to make things and play with robots.

Jameco welcomes the contributions of its customers. Frankly, we think what you write is more interesting than anything we could write. Share your electronic component story, project, or challenge, and we'll share it with the world. Send your story to

MyStory@Jameco.com