

Trilife: A kit for large group workshops

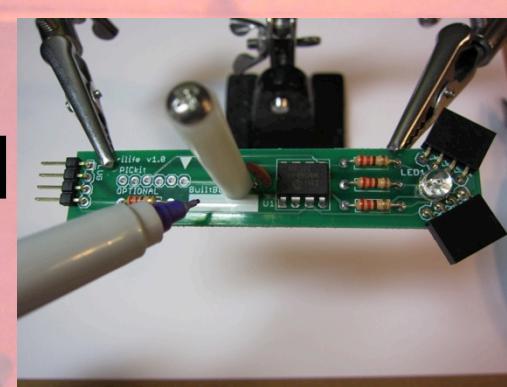
Cindy Harnett, University of Louisville

http://wiki.lvl1.org/Trilife_Workshop

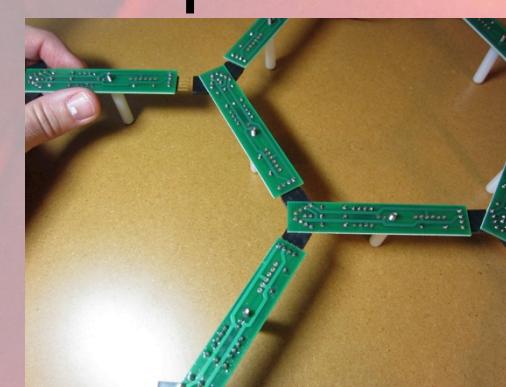
What this is

The open source Trilife kit is designed for teaching large groups about circuits, soldering, and collaboration. It is hosted at the LVL1 hackerspace in Louisville.

Each Trilife board



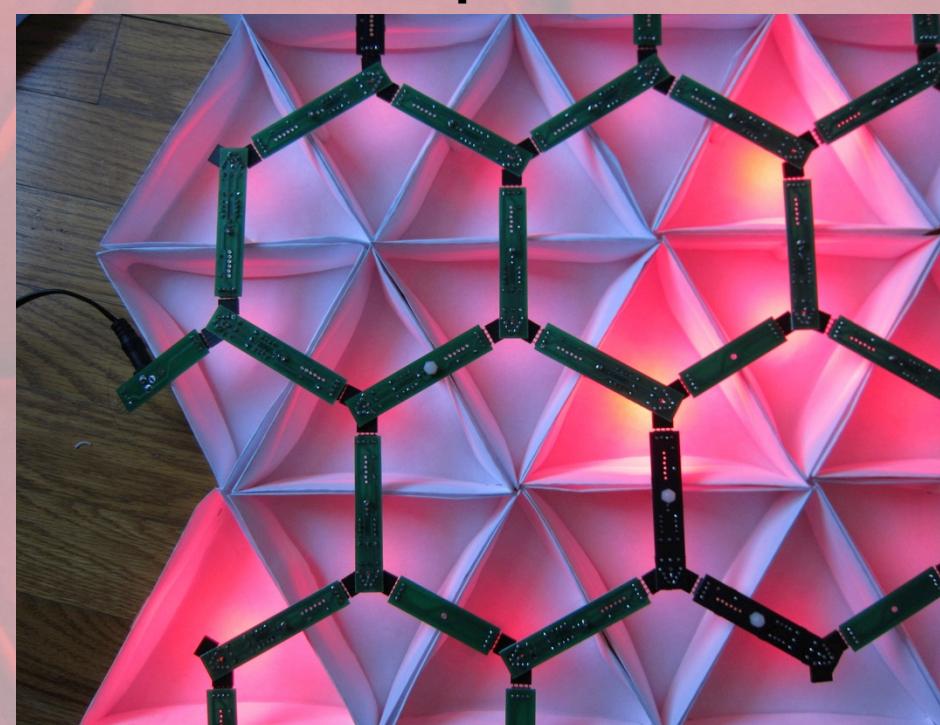
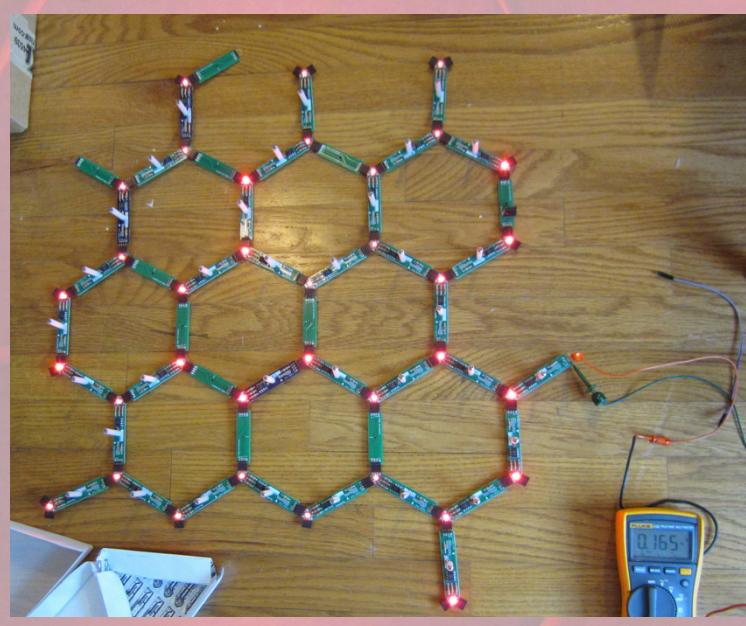
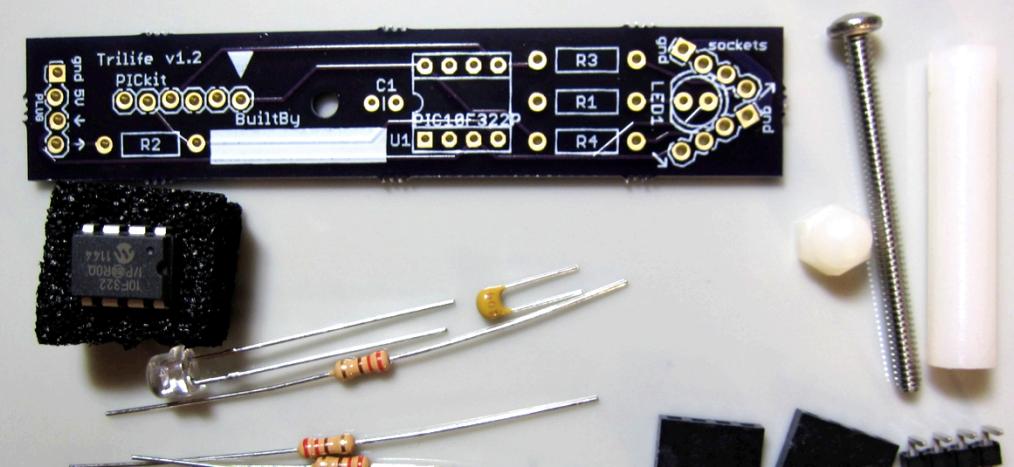
connects to 3 neighbors



If one or more of your neighbors' lights is fully on, yours goes on a bit later. Then it fades off. This simple rule creates moving patterns across the array. There is no central processor or clock—each board controls one light.

Building it

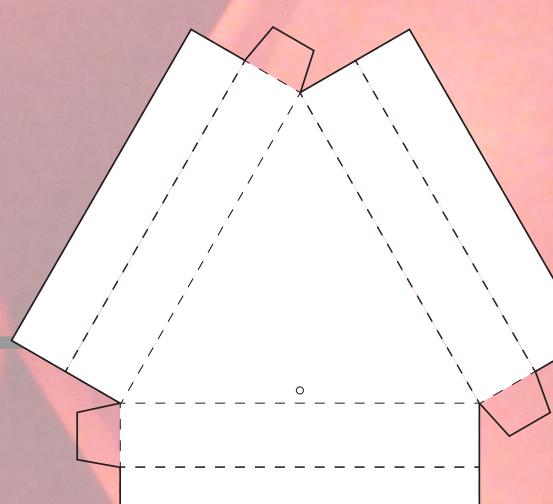
The 10-component Trilife kit is based on the Microchip PIC10F322 processor.



For every 2 boards (on average) in a hexagonal array, build 1 connector board



Cut and fold light diffusers from vellum or office paper



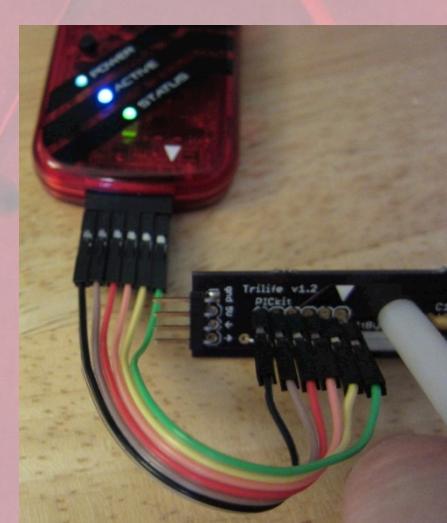
Connect any board to 5V; each board draws 5-10 mA

Random patterns are triggered at startup—if a pattern dies out it tries again
The bigger the array, the more likely a pattern will “catch” on the first try

Questions

Can you reprogram it?

Yes, you can reprogram it in circuit using the PICkit 3 and a 6-pin header. You do have to disconnect each board from the array.



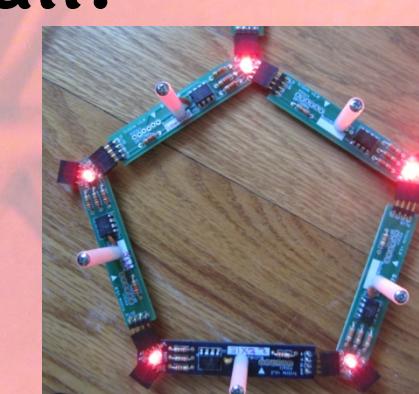
Could you connect them in 3D and make a light cube? You would have to change something major (software, hardware) to deal with 6 neighbors. The microcontroller has 4 I/O pins—one controls the LED and the other three count the frequency of pulses from the neighbors.

Do I have to create closed hexagons?

No, but there should be at least one closed loop for cyclic patterns.

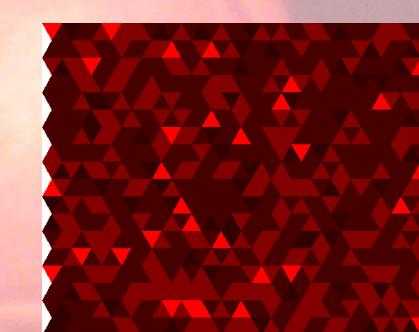
Could you make a buckyball?

Probably, you need some pentagons and this doesn't seem to break things



What happens at edge of the array?

An edge board never detects a lit neighbor on that side.



Can you simulate this?

From the customizable Cellular Automata simulator at www.jcasim.de here are 800 simulated Trilife boards!

