Christmas Tree PCB

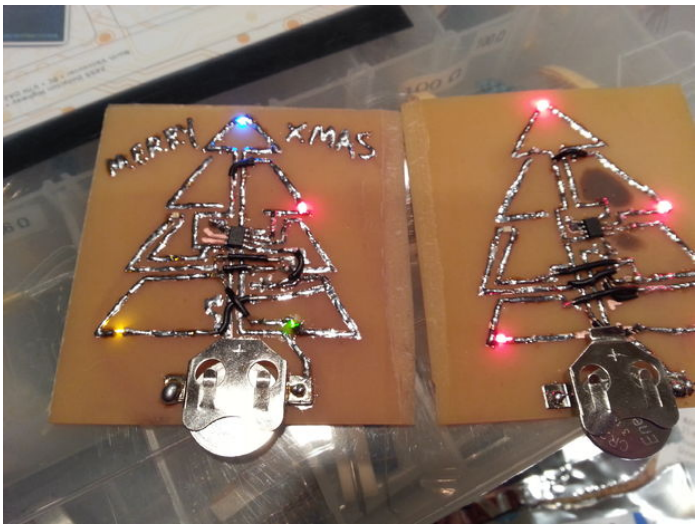
by LostRite on December 16, 2012

Table of Contents

Intro: Christmas Tree PCB

This is a fun way to light up a Christmas card. It will help if you have some experience in any of the following areas: chemistry, PCB etching, avr programming, writing simple C programs, and / or soldering (surface mount soldering). If you don't then the trickiest part will be the soldering since it is tougher than through hole components.

If you do know what you are doing, it only takes half an hour (tops) to finish each card. Here we go!



Step 1: PCB etching

there are plenty of great PCB etching tutorials out there, so I will be brief about the method I used. If you aren't familiar with this process and want to try etching your own board, it is a good idea to learn the procedure and safety involved first.

You will want to be in a well ventilated area (preferably outside) and try not to breathe the gases or let any spills find your skin or clothes... with that said, it isn't the end of the world or your life if that does happen. Just wash thoroughly with water for a couple mins. I personally use a well ventilated indoor room (a kitchen would work) with a sink nearby (a porcelain sink, not metal so it doesn't etch the sink).

Materials / equipment:

- pyrex baking dish
- acetone
- HCl (muriatic acid)
- hydrogen Peroxide (3%)
- stir plate [optional]
- sink
- needlenose pliers or tongs or something similar
- Cu coated fiberglass board with pattern on

The chemicals used are 2 parts hydrogen peroxide (3%) and 1 part muriatic acid (I think 68% HCl? whatever you find at most hardware stores). Fill the pyrex dish up with enough of this 2:1 solution such that it covers your pcb board that you want to etch (getting to this part). Let sit / stir for 5-10 mins and use a pliers to remove. Wash with water. Then get a paper towel and dab on some acetone to wipe off the permanent marker mask. Done!

Step 2: Draw the design

You can draw whatever circuit you want. Here is the one that I made. I kinda just made it up as I went, so there are more efficient ways of drawing it. I knew I wanted a tree shape so I designed it around that.

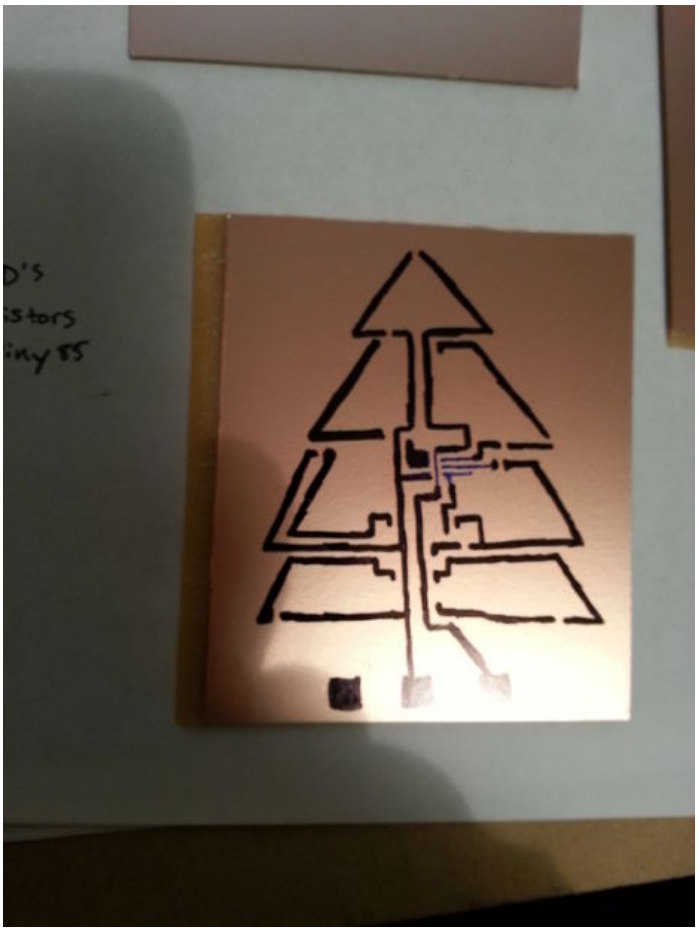
Once you know what you plan on drawing, use permanent marker (sharpie) to draw it on your board. The sharpie will act as a mask during the etching process and can later be cleanly removed with acetone.

The final design that I used included 6 LEDs, 4 resistors, a ATtiny13a, and a coin cell to power it all. A schematic is also included in the .zip file attached.

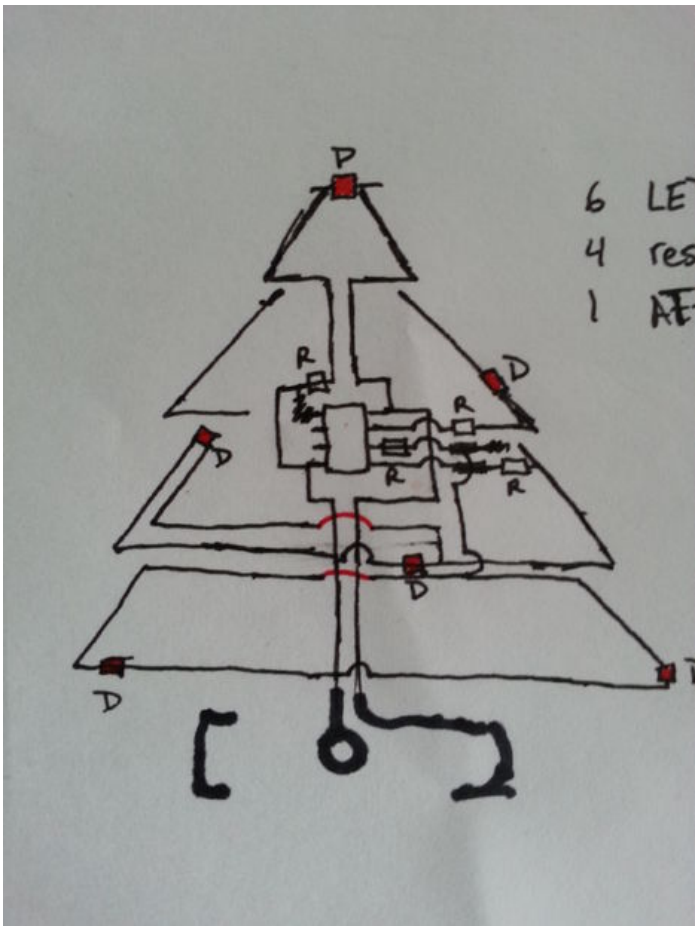
Components List: (all components are surface mount)

- 100 ohm resistors (mouser)
- ATtiny13a (mouser)
- Red LED (mouser)
- Blue LED (mouser)
- Yellow LED (mouser)
- Green LED (mouser)
- White LED (mouser)
- Coin Cell Holder (mouser)

*Note: These components are larger than the ones I used because I thought that the ones I used were too small. So the LEDs listed above are all 0603 or larger (I used 0402 parts).




MERRY
XMAS

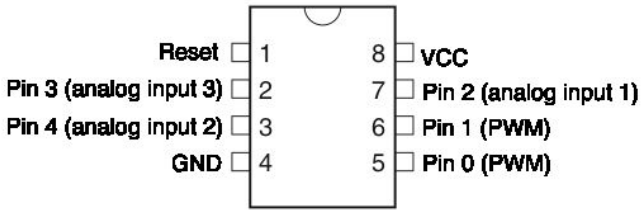

P
R
D
R
R
R
D
D
D

Step 3: Soldering

This is the toughest step just because it is so hard to work with such small parts. I despise myself for using 0402 components. I will never do that again (hopefully). The 0603 SMD (surface mount device) components were just fine to work with though. To be even safer, I would recommend the 0804 parts. See the previous step for a parts listing.

Make sure you know which side of the LED connects to - and to +. For SMD LEDs, there is a small green dot on the top of the LED that indicates that side connects to the - lead. There is also an arrow generally on the bottom that points from the + to the - side. It doesn't matter for the resistors. Also, if your SMD ATtiny13a doesn't come with a dot on it to denote pin 1, figure out which way the label reads (Atmel, etc). Whatever it says, you just need to know the direction words go in. Pin 1 will be the one below the first letter (bottom left as you are reading it). It can be tough with these small guys if you don't have a magnifying glass.

Make sure to be extra careful with the microcontroller not to apply to much heat and ruin it. Also, be careful not to connect pins to each other.



Step 4: Prepare for Programming

In order to program the ATtiny we will need to attach wires from the microcontroller:

- Reset (pin 1)
- ground (pin 4)
- Vcc (pin 8)
- SCK (pin 7)
- MISO (pin 6)
- MOSI (pin 5)

These wires will be temporary, so attach them wherever you can to the circuit. It is probably safer not to solder right onto the microcontroller pins though... Sorry I don't have good pictures of this. It was hard to take pictures when my camera wasn't focusing well up close.

You will need an AVR programmer (again, I used the one from adafruit, USBtiny). If you are using such a programmer, there are either (or both) a 6-pin and a 10-pin ISP connector. You can use either, but you only need 6 of the wires. Those being the same 6 listed above. I have included a picture of the layout of each of these ISP connectors for reference so you know how to wire them up to the circuit board. I would suggest using a target board or breadboard or just stray wires will work fine if you just want to jam them into the connector.

Also, if you are using your programmer to supply power to the chip, you will want to bring down the 5V that it supplies down to something closer to 2.8V so that you don't blow the LED on the top. To do this, just attach a resistor (~ 240 ohm) in series to either the Vcc or ground lines between the programmer and the microcontroller / pcb.

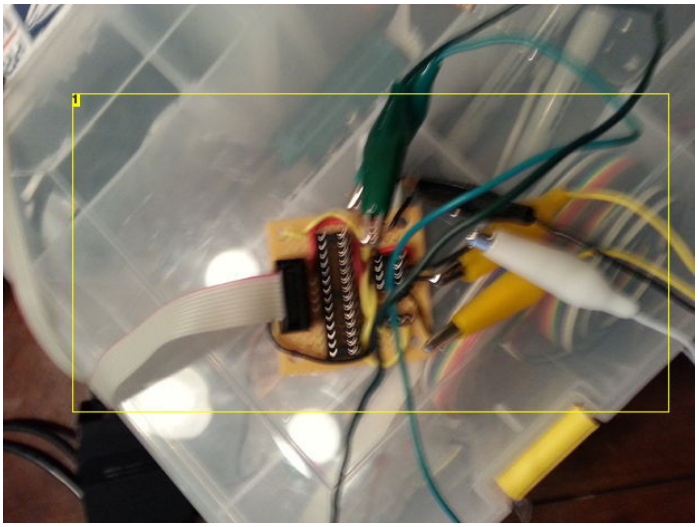
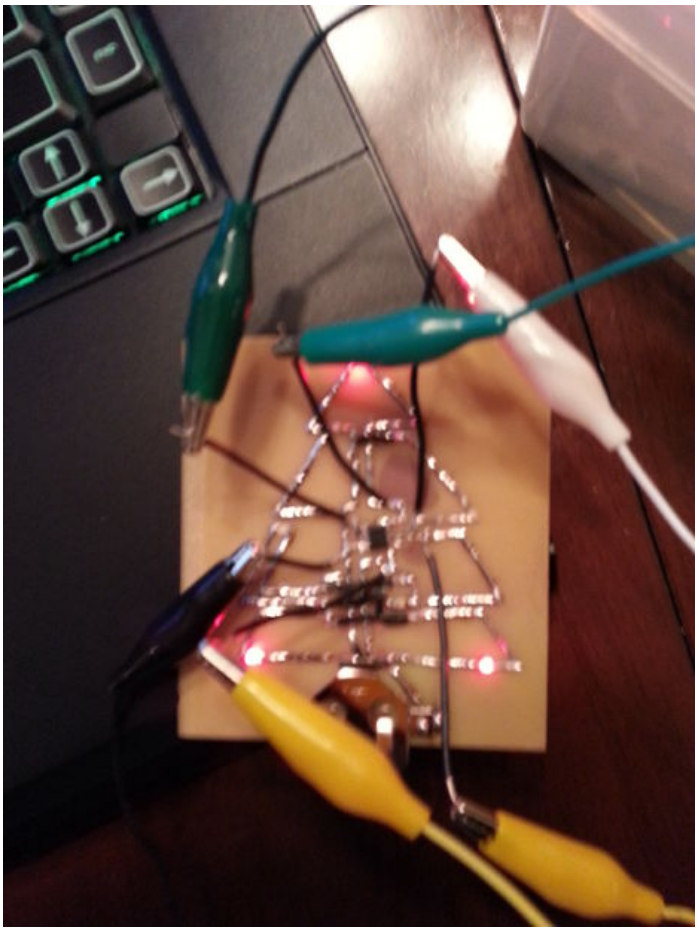
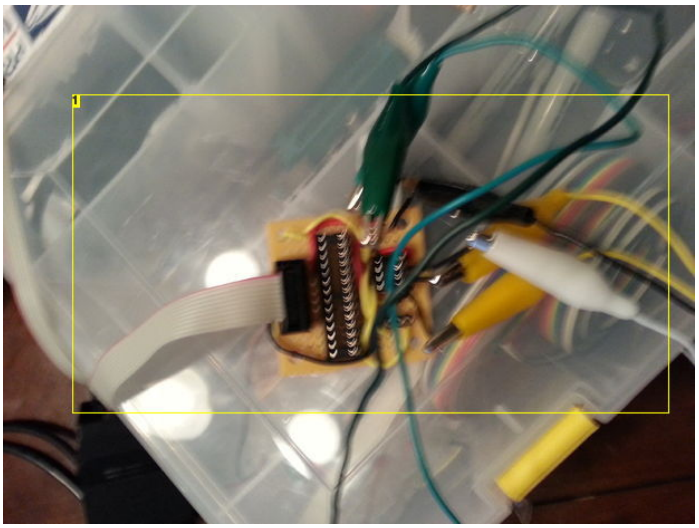


Image Notes
1. my target board that I use for stuff like this...





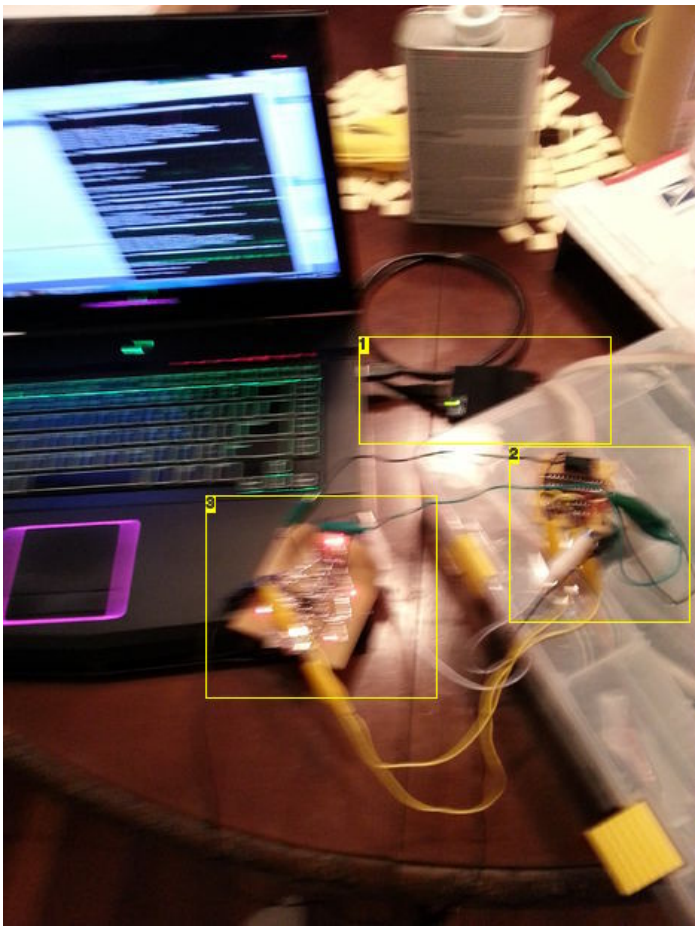


Image Notes

1. USBtiny
2. Target board
3. PCB w/ microcontroller pins connected to target board

ISP Connectors: 6-pin & 10-pin

Step 5: Programming & Code

I used a programmer like the USBtiny . Feel free to use whatever you have available. I believe you can use an arduino as ISP if you don't have an AVR programmer handy. I didn't look into arduino code or even if the arduino can program attiny13's. I would imagine it can just fine though.

Hook up the programmer to match the configuration in the previous step.

You will need the .hex file (I attached it in the .zip file so you can download it). Or you can also compile it yourself if you want to modify the code. If you don't know what software to use, you can always use the free Atmel Studio 6 software.

The code basically uses XOR to flip / flop the state of the LED. It chooses which LED to flip based on a random number modulus 3. This way LEDs can be on at the same time and each have their own random chance to stay on. It makes a flip / flop decision every x seconds where x is another random number. I know it is pointless, but it also scrambles the seed everytime it loops. So it still uses the same random number everytime it gets turned on. A better way would be to read an input off of pin 3 or 2 (PB3 or PB4, respectively).

The code is pretty straightforward:

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>

int main(void)
{

DDRB = 1<<PB0;
DDRB |= 1<<PB1;
DDRB |= 1<<PB2;
PORTB = 0;
int n;
while(1)
{

n=((n+57)*13)%10057;
srand(n);

int r = rand()%3;
if (r==0) PORTB ^= 1<<PB0;
if (r==1) PORTB ^= 1<<PB1;
if (r==2) PORTB ^= 1<<PB2;

int r3 = rand()%10;
for (int t=0;t<r3;t++) { _delay_ms(75); }

}
return 0;

}
```

File Downloads

xmas_card.zip (21 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'xmas_card.zip']

Step 6: Uploading code

To upload the code, I used AVRdude. I recommend you do as well :) There is a fantastic tutorial on this by ladyada on the adafruit website. Look it up to learn the basics.

Assuming you have downloaded what you need for avrdude, lets open up a command prompt window (for you windows users that is). If you are using the USBtinyISP that adafruit sells, use the code exactly as I have it (skip down), else you may have to find out what your programmer is called. To figure out what avrdude calls your programmer, type in

avrdude -c ?
The "?" can really be any made up word, it will spit out a list of options.

Once you have your programmer found, you will also need to know what microcontroller you are using. If you are using the attiny13 like I am, use the code I have, otherwise you will have to find it by searching the list generated by typing in:

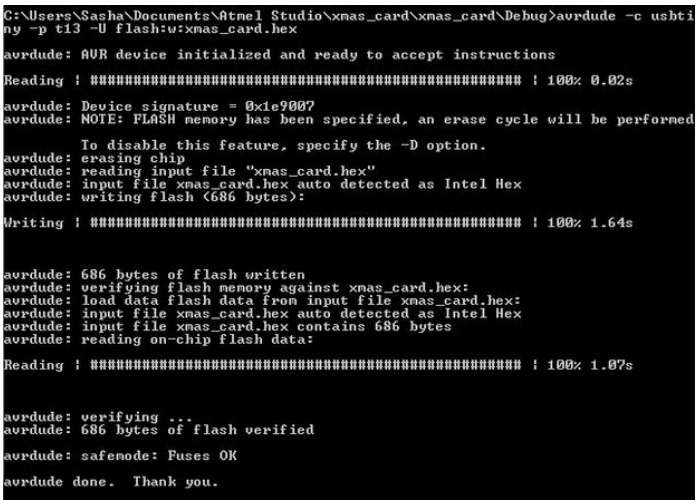
avrdude -p ?
Once you have that, you will need to figure out where your file is located (what directory it is in). I always find it is easiest to navigate to the folder where the .hex file is located before typing in all the avrdude stuff.

To navigate to where your .hex file is, use the change director command "**cd**". For example, lets say you were in the directory "C:\Users\" and you wanted to get to the folder Documents (which is in the folder "Sasha" [my name] which is in Users), you could either use one command "cd Sasha\Documents" or use two consecutive commands "cd Sasha" + "cd Documents" to get there. I'll let you figure it out from there. It is helpful to know the command "ls" which lists what files and folders are in your current directory. Once you have navigated to the directory with xmas_card.hex you can proceed to the avrdude command that comes next. You can see the directory for my computer in the top line of the picture on this step.

If you have names the .hex file something else, you should change your command to match that. Assuming you are in the right directory and you have done everything above, and have plugged in the programmer properly, execute the following command:

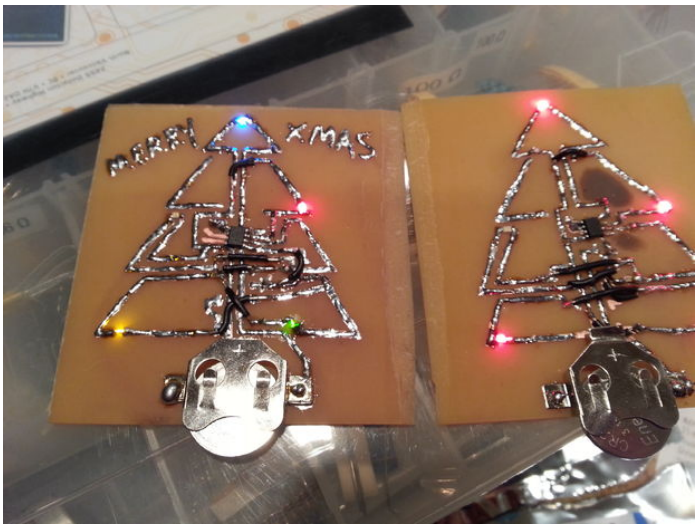
avrdude -c usbtiny -p t13 -U flash:w:xmas_card.hex
if done properly, you should get a screen like the one shown in the picture. If you get errors and need help, post them in the comments or send me a message.

You may now desolder the wires we connected for the programmer. This is fairly straightforward I believe.



Step 7: Plug and Play

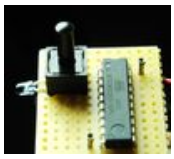
All you have left to do is insert a coin cell CR2032 battery into the holder and watch it sparkle!



Related Instructables

Ambient Light Gift Badge by frickelkram

Hackable Christmas card & ornament by ian



Infra-red Remote Intervalometer for Nikon Cameras by jwagnerhki

LED Hanukkah Menorah by barney_1

USB PCB Business Card by frank26080115

Installing Attiny13 core files (Photos) by diy_bloke