

Mail notifier clone

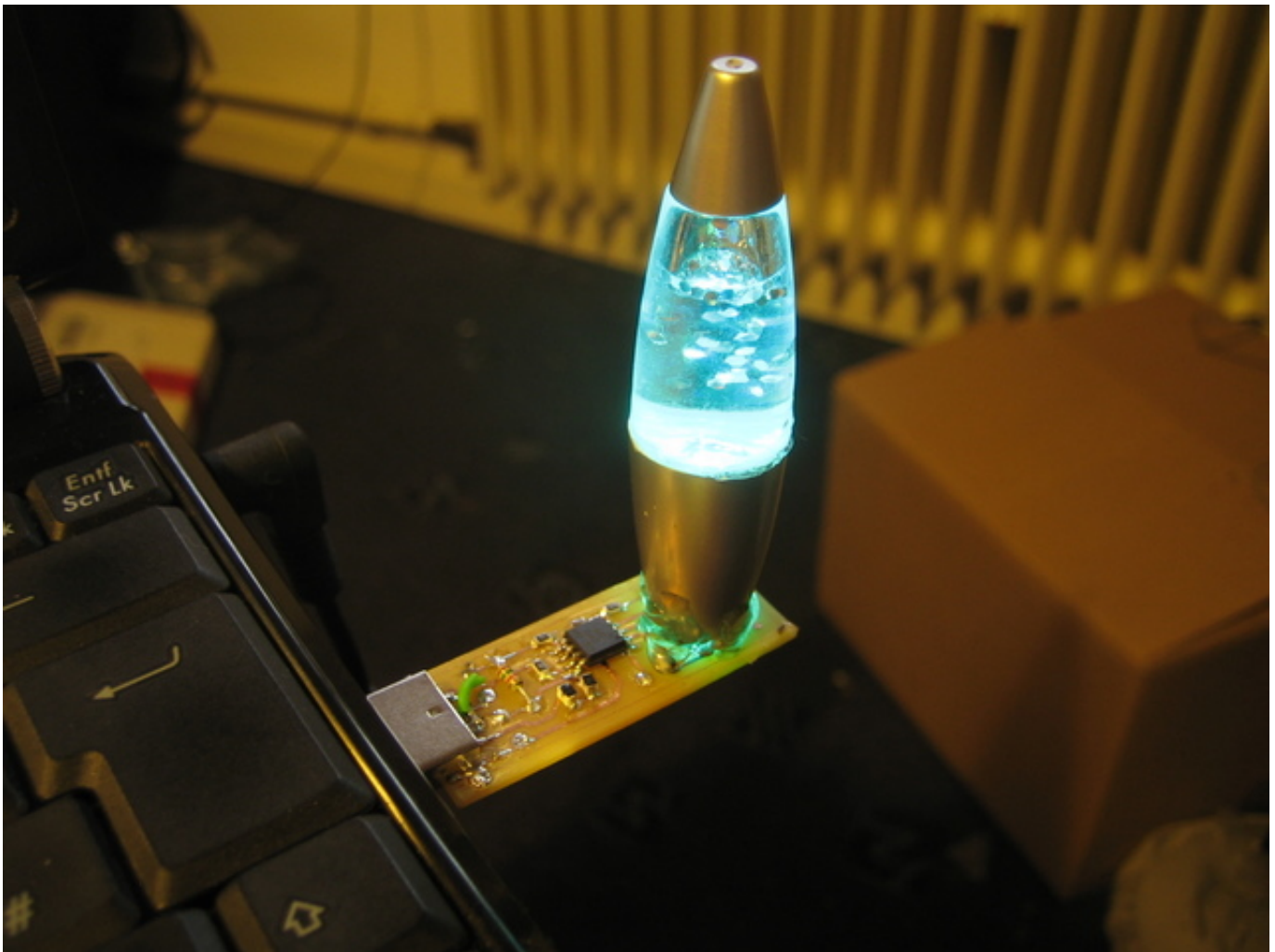
June 29, 2009, 10:41 pm

Filed under: [ATTiny45](#), [AVR](#), [build](#), [Electronics](#)

I've been a bit slack with blog posts of late. I'll probably start with more electronics when the weather turns, but for now I'm enjoying the sun.

In the mean time several of the forum members on [eeeuser.com](#) have been attempting to build a mail notifier inspired the LED [cube](#) that I've built. They've put their progress in this [thread](#). Forum member posted some pictures to that thread. They've gone a bit further than I have by etching their own PCBs and using SMT components.

Here is the completed circuit controlling a mini lava lamp from forum member [AlphaCentauri](#) whom has also adapted with [Twitter/ICO/Mail notifier software](#), [dBird](#) to control the cube.



[9 Comments](#)

Cube follow up

April 23, 2009, 10:37 pm

Filed under: [ATTiny45](#), [AVR](#), [Electronics](#) | Tags: [attiny](#), [ATTiny45](#), [AVR](#), [code](#), [prototype](#), [usb](#), [usbtiny](#)

I've been asked a few more questions about the Cube, so here is a follow up post with some answers.

Here are the previous posts with the information so far:

- [Initial prototype and bread board version](#)
- [Firmware and Software](#)
- [Schematic and final build](#)

Questions

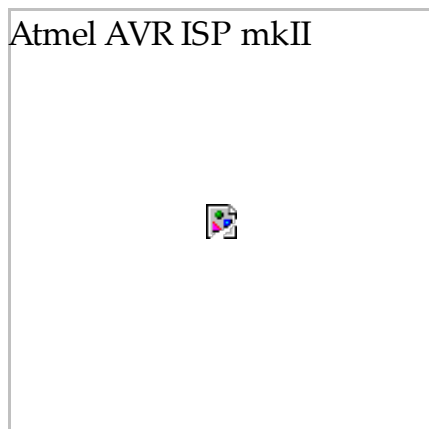
What toolchain did you use?

I use [Crosspack tool chain \(formerly AVR MacPack\)](#)

I also use [V-USB \(formerly AVR-USB\)](#) by the same author as Crosspack. This is a software USB implementation that works great in this scenario when only a small amount of information needs to be transferred.

How do you flash the chip?

I've created a small board powered by USB which I can insert the ATTiny into and attach an Atmel ISP mkII. Flashing the program is as simple as typing 'make program' from the firmware directory in the code.



How much did the components cost?

- ATTiny45 £1.26
- USB Connector £0.29
- LEDs approx £0.60
- Resistors, Diodes, strip board approx £0.60

The cube itself was a freebie and I could have salvaged the LEDs. I've deliberately aimed to keep things simple with as few components as possible. It has cost less than £3 (~\$5).

Is it fast enough to do colour fading?

Yes. It uses software PWM to create 24-bit colours. That means the LEDs are turning on and off really quickly to give the impression that they're either bright or dim and mix in different proportions to create all the shades.

At the moment I just run the executable (in the commandline directory in the code) many times to create the fade. It wouldn't be too complicated to extend the firmware to add smooth fading or alternatively to the command line tool.

How fast can you change the colours?

Pretty much instantly. However, it depends what you mean by 'change'. As I mentioned before the LEDs are turning on and off constantly. You are limited by the transfer rate of the USB, but you could add colour fading in the firmware reducing the number of instructions required to change.

Can I see more pictures?

I probably won't revisit this project, but I promise to add more to future posts.

My cube is a bit dim how can I make it brighter?

It depends, I have build one board where I have removed the resistors protecting the LEDs as they are rated at approximately 3v. It is also possible to run LEDs at a higher voltage than their max rating if you reduce the duty cycle as described in this [article](#).

8 Comments

USB caps lock fob

April 3, 2009, 2:22 pm

Filed under: [ATTiny45](#), [AVR](#), [build](#), [Electronics](#) | Tags: [attiny](#), [ATTiny45](#), [AVR](#), [joke](#), [usb](#), [usbtiny](#)
When building the USB mail notifier, I found this [blog post](#) describing how to build a tiny USB device that randomly turns the caps key on. One of my colleagues, Simon, particularly hates accidental caps presses, so much so that, he physically removed the key from his keyboard. This made him the ideal target for this joke.

The hardware required is very similar to the email notified I built. It is actually possible with a reprogram of the chip to use the same device. However I want to make this small enough to be hidden.

As with all my projects I'm trying to avoid making PCBs or using surface mount devices. I'm sticking with stripboard and normal sized devices. I used the same vertical mount of inline components.

I have removed the USB connector from an old memory stick and, although this is surface mount, the pins are spaced far enough apart to sit on a track each.

When all the parts are soldered and the chip is flashed I tested the device. It doesn't seem to work on OS X, but fortunately it does on Windows.



Caps locker fob

As for Simon, he was perplexed for about 40 minutes, but saw the funny side when he figured it out 😊

2 Comments

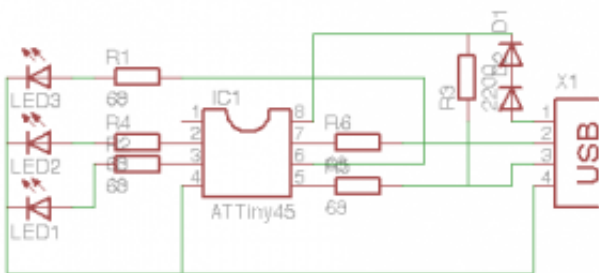
Building the cube

March 30, 2009, 8:18 am

Filed under: [ATTiny45](#), [AVR](#), [Electronics](#), [Prototyping](#) | Tags: [attiny](#), [ATTiny45](#), [AVR](#), [led](#), [pcb](#), [rgb](#), [usb](#), [usbtiny](#)

I've finally set about building the USB mail alert.

In my previous post, I showed the bread board prototype of my cube hack. Here is the adapted circuit diagram.



schematic

NB: 2 is D+ 3 is D-

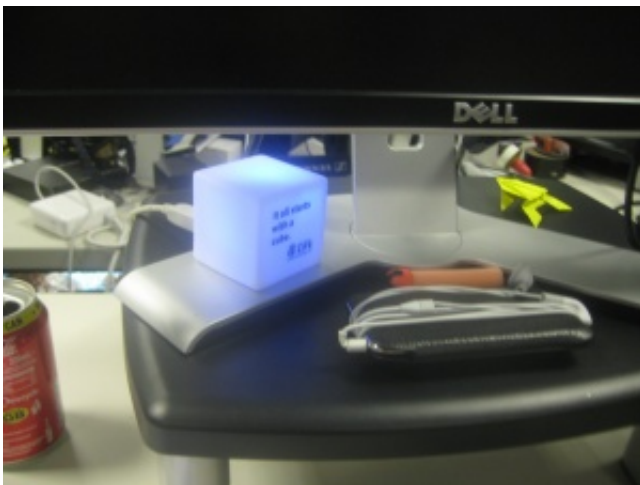
When laying out circuits, it is a good idea to, start with the largest component first, in this case, the ATTiny45. I want to program the chip first so I am using a DIL socket. This also has the advantage of protecting the chip from excessive heat when soldering. I'd also like to use as small a board as possible so I am limiting the width to six tracks. Place all the components on the board before soldering so you can get an idea of how large the board needs to be. I followed the layout of the breadboard, however I want to make the circuit small so I have mounted the resistors and diodes vertically, with one leg bent back.

I also tried to avoid cutting tracks with the spot cutter and instead used a knife to allow components to be closer to each other. This requires a sharp knife and care must be taken to cut only one track. Soldering is also more challenging, solder sparingly, to avoid bridging across your cuts.

I cut the tracks under the chip; note that the reset pin is left disconnected. If you needed extra IO you can flash the chip to use it as such but after that you can no longer reprogram it so I avoid that as I may use the chips from previous devices in future things.

I soldered the LEDs at the end of their leads, towards the end of the PCB. This means they will be in the center of the difuser. They also have a series resistor to reduce the AVR output voltage from 3.6V to the appropriate level. This resistor may not be required with some ultrabright LEDs.

To fit in the cube, I have removed the circuit, the coin cells and cover. The cube is going to be used as a cover for the new circuit so I have just cut a space using a sharp knife allowing space for the LEDs to fit into the diffuser. The circuit is slotted into the empty cube to complete the device. Plug it in and test it using the libusb command line tool, mentioned in a previous post.



The finished cube

19 Comments

Cube Hack part II (software)

March 10, 2009, 10:41 pm

Filed under: [ATTiny45](#), [AVR](#), [Electronics](#), [Firmware](#), [Software](#), [Uncategorized](#) | Tags: [ATTiny45](#), [c](#), [code](#), [led](#)

I've created a Google code project with the snappy title of [avr-usb-rgb-led](#). It is based on the hid-custom-rq example from AVR usb examples. I've added 3 new commands, one for each LED. It uses a software PWM to control the brightness of each LED based on the value set.

There is a corresponding program using libusb which allows you to send the commands, it simply allows you to set the colour of the cube with an RGB tuple on the command line.

To get the code you need to check out using subversion. To flash the chip you probably want to use the makefile.

6 Comments

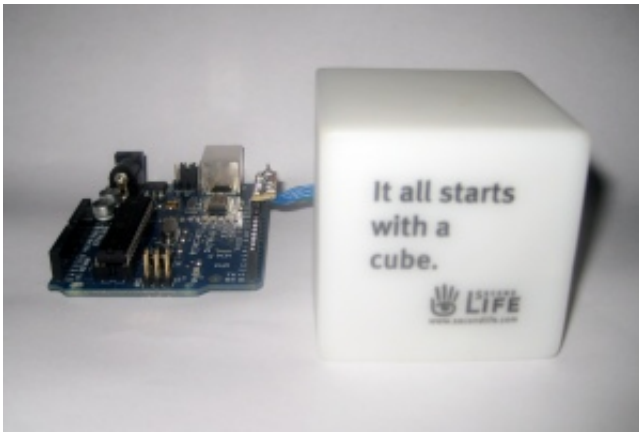
Glowing Cube Hack

March 4, 2009, 10:05 pm

Filed under: [Arduino](#), [ATTiny45](#), [Electronics](#), [Prototyping](#) | Tags: [Arduino](#), [attiny](#), [ATTiny45](#), [AVR](#), [hack](#), [hardware](#), [rgb](#), [usb](#), [usbtiny](#)

One of the promotional things that Linden Lab gives away is a small colour changing, glowing cube. Last year, we attended a job fair where we gave several away. One of the people who received one decided to hack his cube and made a [Gmail notifier](#).

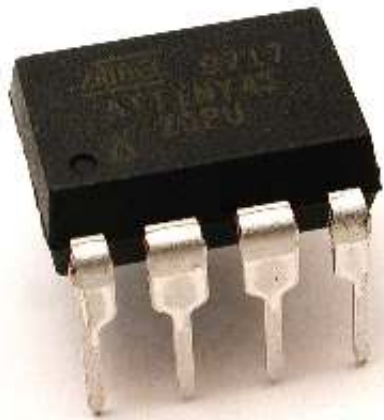
I decided to copy the hack, bought myself an Arduino, and set about hacking the cube. I went for a slight improvement where I individually wired each LED. This means that using the PWM from the Arduino I was able to set the colour of the cube to whatever I chose.



Arduino and Cube

The main problem with this hack was the physical size of the Arduino. It's too large to fit inside the cube. I could have attempted to use one of the smaller Arduino variants, but it would have been a bit pricey, costing about £30 for the Mini and the USB adapter.

I'd recently discovered [AVR USB](#), which implements, USB in software, for any AVR microcontroller. USB needs two wires and I needed another 3 to control my LEDs, so even the smallest microcontroller would be enough and it doesn't even need a crystal. AVR USB needs at least 2K of flash memory. There is an example project, [EasyLogger](#), that uses ATTiny45 so that's what I'll use too.



(C) HWTech, 2007

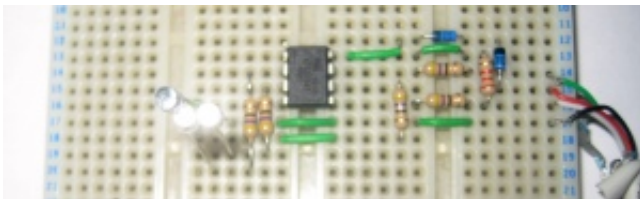
ATTiny45

The EasyLogger project is a good starting point, but I'm going to make some changes. First, I want to control the LEDs, so I replace the input voltage and start switch with a resistor LED pair.

USB uses 2.8-3.6V for high on the data pair but provides 5V for power. Since I'm going to be powering the ATTiny from the USB I need a way to make the outputs deliver the lower voltage. The EasyLogger project uses Zener diodes to provide level conversion, but unfortunately, I don't have any Zener diodes.

The AVR USB provides an alternative suggestions of using a 3.3V voltage regulator for reducing the power (which again I don't have any of) or a cheaper alternative, using a pair of normal diodes. Each diode has a drop of 0.7V so the pair drops the power voltage to 3.6V.

Here is the first prototype circuit on my bread board.



Breadboard prototype

I'll describe the code for the firmware and driver in later blog posts.

6 Comments

[Blog at WordPress.com.](#)

Theme: [Benevolence](#) by [Theron Parlin](#).

Syndicate entries using [RSS \(Really Simple Syndication\)](#) and [Comments \(RSS\)](#).