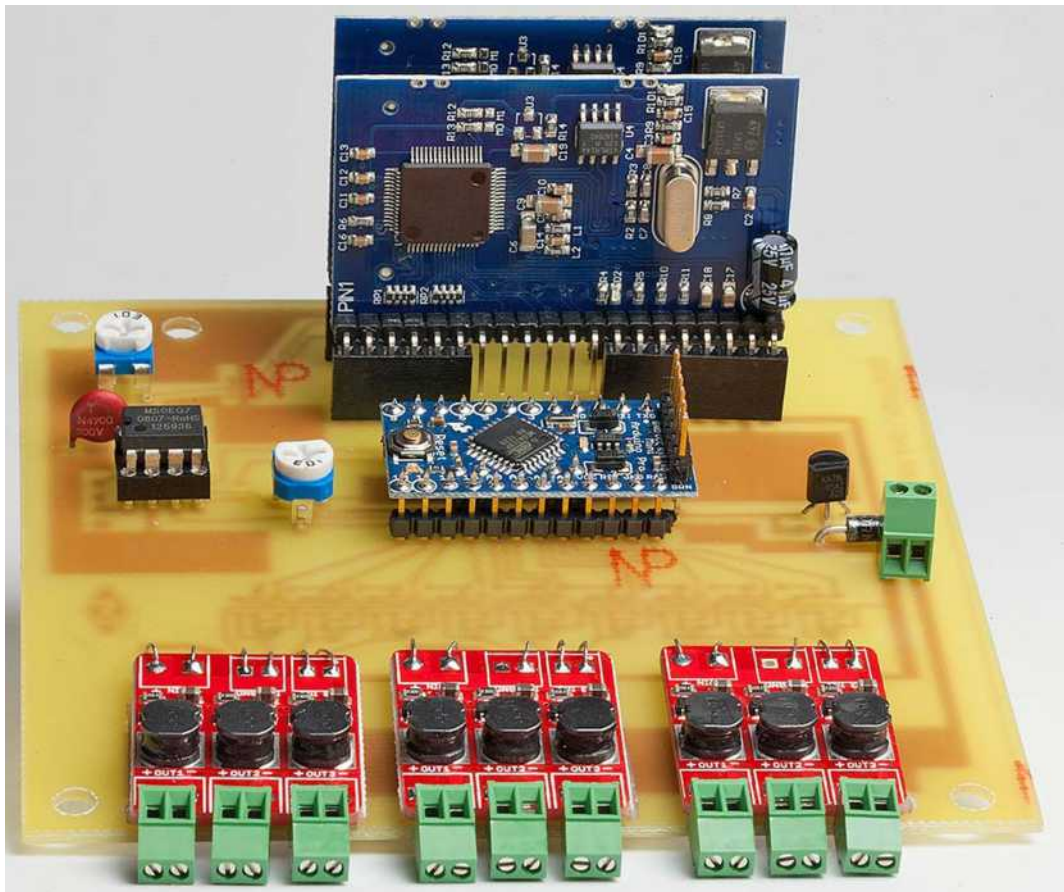


Tres Amigos Labs Thunderstorm Generator

Bob #1 February 14, 2016, 10:34pm

In the build-out push for David's September 2013 layout tour we installed dimmable LED strip lighting in the rain room, a/k/a "where the sun don't shine", the town formerly known as Darwin, now Linnwood. The build-out also calls for lightning effects and a thunderstorm sound generator.

For the latter we have a work-in-progress, the Tres Amigos Thunderstorm Generator.



This board drives 9 digital camera flash LEDs by way of 3 Sparkfun 350 mA high-current LED drivers (red boards along the bottom.) LEDs are deployed in a horseshoe shape around the room.

Logic is controlled by an Arduino Pro Mini (center) and audio provided by a pair of MP3 player boards from MDFLY (actually they are the TENDA TDB380 MP3 Module.) The whole thing runs off a 12 Volt, 2 Amp regulated wall wart.

Currently the LED lighting operation works fine. I'm currently working to get the MP3 players going

(documentation is horrible and TENDA engineering not up to my expectations.)

up148 #2 February 14, 2016, 10:34pm

Outstanding! Your fabricating abilities never cease to amaze me Bob. What an incredibly talented group of modelers in one location. Must be that good clean mountain air.

Butch

Bob #3 February 14, 2016, 10:35pm

I've been remiss in following-up on this project. It has been finished and installed for a long time since the initial post.

A few more details for the Arduino savvy are in order. First off, only a single MP3 player was installed, as a second one to play really big flash/bangs didn't fit well into the soundscape. I loaded a single hour long thunderstorm sequence that proved suitable.

The software has a learning mode, during which it "listens" to the audio, and saves a list of major thunder events and their statistics. For analysis it uses a MSGEQ7 graphic equalizer display chip in the 8-pin DIP package next to the red ceramic capacitor in the photo above. This part, from Sparkfun, was designed to separate the audio spectrum into 7 frequency bands and output an analog voltage corresponding to the energy in each of those bands. During learning playback, the Arduino reads this analog voltage and measures thunder "rumble" as a weighted combination of the two lowest frequency bands. When the statistics exceed a threshold, it starts logging a thunder event with time since start of playback. At the end of the recording, the Arduino writes the thunder event log into the EEPROM for subsequent playback. With the current sound file, 180 thunder events were logged over about 1 hour, an average of every 20 seconds but spaced farther apart in quiet times and closer together during heavy rain.

During normal operation, the Arduino starts the MP3 player then browses through the thunder log. The Arduino controls lightning LED turn-on time, how many LEDs, LED intensity (through PWM) and turn off time. It calculates how far in advance of a thunder clap to fire the lightning stroke based upon recorded statistics. Quieter rumbles may flash 5 seconds before the thunder can be heard, and briefly illuminate only one LED. The loudest thunder clap fires all 9 LEDs at full-intensity and for a long duration, with multiple strokes, starting only one half second before the clap can be heard.

During quiet periods, the code reads ahead through the log and knows that there will be a long time period until the next scheduled lightning event. During these interludes it tosses-in random dim flashes, barely visible during dimmed daytime operations, but plainly visible during night operations.

After paying close attention during the 2013 thunderstorm season in Colorado, I intend to update the playback code to use longer lightning flashes.

The audio jack on the PC board feeds an inexpensive Lepai 2x15 watt class D amplifier from Parts Express which drives some large 1970's vintage stereo speakers that David saved. During guest visits the volume tends to creep up towards 11, but during normal operations it should be *far* more restrained.

The following vintage photo (by now!) taken in September, 2013 with a cell phone camera shows the current installation.



A few more notes for the Arduino community. I procured the MP3 players from MDFLY.com. The main ASIC has had its markings laser or sand blast removed. It may be the same as a TENDA module. I found both sets of vendor documentation to be helpful. For the price I still feel it is an excellent choice for my projects despite some limitations. For an inexperienced programmer the limitations might feel overwhelming. "Your mileage may vary." The SD card reader implementation on my boards completely ignores file names. It reads the *location* of the first entry in the FAT directory as the first audio file, etc. Format the card before using! Then copy the file you want to be the first to the card, then the second...