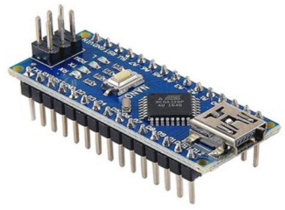
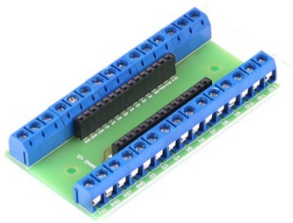
Chapter 9 – Adding Audio to an Arduino

April, 2020
David Ackmann – Gateway Division NMRA

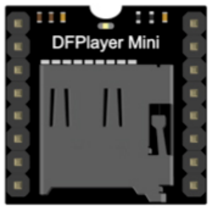
With the proper hardware, an Arduino can play MP3 files stored on a micro SD card, such as are commonly found in cameras and cell phones. This tutorial explains how this can be accomplished.

I usually use an Arduino Nano with an expansion board, plus an external power supply, MP3 player board and small micro SD card (I use a DFPlayer Mini, AMAZON ASIN: B07JGWMPTF), a speaker (AMAZON ASIN: B0738NLFTG), a voltage regulator (AMAZON ASIN: B07WQJ2GD6), amplifier (AMAZON ASIN: B01MYTZGYM), barrier strips, spade connectors and jumper wires.


Arduino Nano

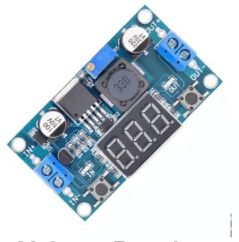

Nano Expansion Board

Power Supply



MP3 Player

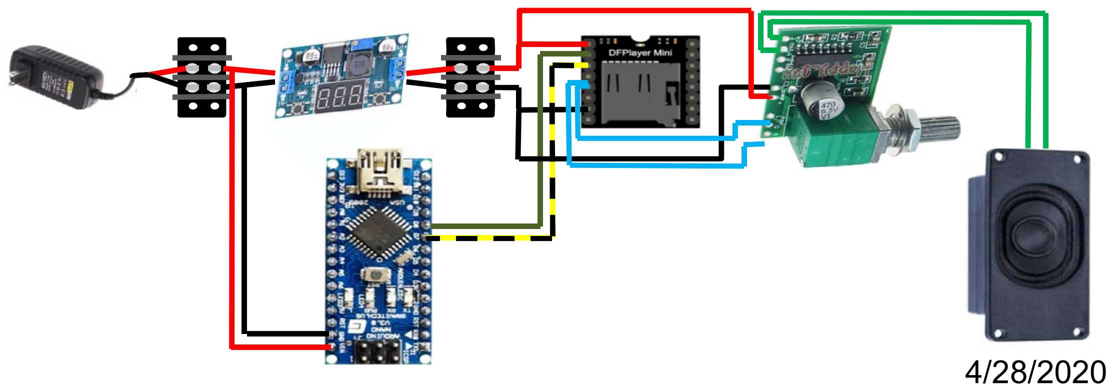
Speaker


Voltage Regulator

Amplifier

Barrier Strip

While you can control the volume level through software, I prefer to use a small amplifier with a knob to adjust the volume. For all my Nano projects I also a Nano Expansion Board because it makes connecting components easier (AMAZON ASIN: B07MGVC18K). The diagram below shows the basic component connections.



4/28/2020 4:11 PM

The 12VDC power supply is connected to a barrier strip (red and black wires), and from there to a Buck Converter (voltage regulator) and to the VIN (Voltage Input) and GND (Ground) pins of the Arduino (make sure you get the polarities correct). Run wires from the output of the Buck Converter to the second barrier strip. Set the Buck Converter to 4.2V before hooking up power from the second barrier strip to the MP3 player and the amplifier. Run wires from pins 7 and 8 of the Arduino (green and yellow/black) to pins 2 and 3 of the MP3 player (it doesn't matter which pins on the Arduino go to which pins on the MP3 player). Run wires from pins 4 and 5 of the MP3 Player (blue) to pins 1 and 2 of the amplifier. Note that the amplifier does not have pins in positions 4 and 7. Run wires from the amplifier pin positions 10 and 11 (green) to the speaker.

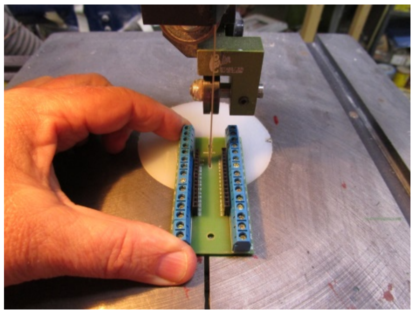
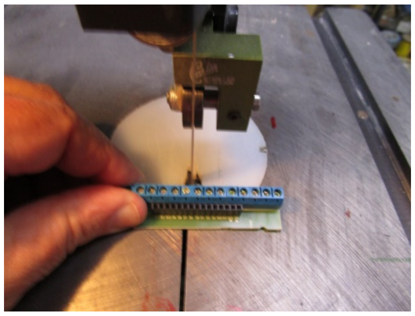
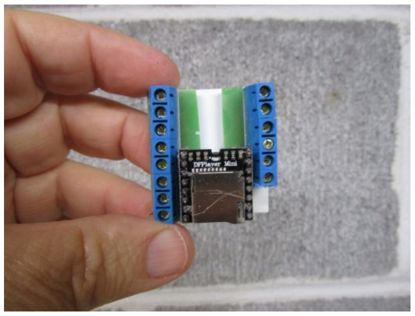
It may seem a bit odd that we need a Buck Converter (voltage regulator) in addition to the external power supply. But this is necessary because the MP3 Player circuit likes to runs on about 4.2 volts, nothing over 5VDC, whereas the Nano will run on anything from about 7 to 12VDC. There are less expensive voltage regulators on the market, but I like this one because it has a display making it easier to dial in the proper voltage, and it runs cooler than a simple 3 pin voltage regulator. Deviate from the recommendation at your own risk.

I like to use barrier strips between the power supply and the Buck Converter/Nano, and also between the Buck Converter and the MP3 Player/Amplifier. It just makes connections easier. You don't have to use spade connectors at the end of your jumpers when attaching them to the barrier strips (you could just tin the wire ends and wrap the ends around the screws), but spade connectors make for more reliable connections.

Before we start connecting components, let's discuss where we find model railroad sounds, and how we edit the sound files we find. As you might expect, we start by doing a web search on "model railroad sounds", or you can visit http:// http://soundbible.com/tags-train.html for starters. Check out their "Royalty Free Sounds" and their "Sound Effects" tab, but if you need something specific, use the site's "Search" feature. Follow the instructions to download a MP3 file to your local PC; often you can just "right click: on a sound file and then on "Save File As" to place it on your PC wherever you choose. You may also consider doing a web search on "sound effects"; not all you may find are free, but if you can find exactly what you want, it might be worth a few dollars. When you find a sound file or files you like, download them to a folder on your PC, then move one or more to your SD card (you don't need a very large card to store a lot of sounds; I often use cards as small as 8GB, but even smaller would work, if you can find them). But be advised that placing a lot of sound files on an SD can be tricky, because the MP3 player references sound files by their position on the SD card, not by their file name.

If you are lucky, you may find exactly what you want. But for my diorama of a lumberjack chopping down a tree, I needed to do some sound editing. "Audacity" is free sound editing software, so I downloaded it and like it. The software has a small learning curve, but it can trim the files to meet my exact length needs, and to adjust their volume. Audacity has many other sound editing features as well, and it now is my preferred tool for sound editing. Visit http://www.audacityteam.org and download a copy; there are some good Audacity tutorials available on YouTube as well.

If you have read “Part 7 – Why an Arduino Nano Instead of an Uno”, you know that I prefer a Nano because the available Expansion Board (AMAZON ASIN: B07MGVC18K) makes it easier to connect jumper wires to the Arduino. I also discovered that I can take the same expansion board, cut it down the middle on a band saw, cut it again perpendicular to the first cut (making the cut at the ninth connection point), then placing the MP3 player chip into the two pieces to make an custom expansion board for the MP3 player, screw terminals and the whole nine yards! I hot glued the expansion board pieces onto a piece of styrene to make it easier to mount on a fascia board. One piece of 0.040” thick styrene should do the trick. This makes connecting the player to the Arduino and amplifier easier and more secure (and because the MP3 player doesn’t need to use pins 9-15, we only need connection points for pins 1-8 and sometimes pin16). Neat!

But be careful! Power tools like band saws can be intimidating and dangerous if you are not experienced, especially when you saw through the metal pieces that make up pin 9, or any other metal part of the original expansion board. Always follow basic safety instructions and wear eye protection. If you feel uncomfortable making these cuts, then don’t attempt them and find someone with a bit more experience to help; there is no shame in staying safe.

I use the other half of the remaining cutoff to build an expansion board for the amplifier. Just 11 pins are needed here, so I made the crosscut at the 12th pin. Again, “Safety First”.

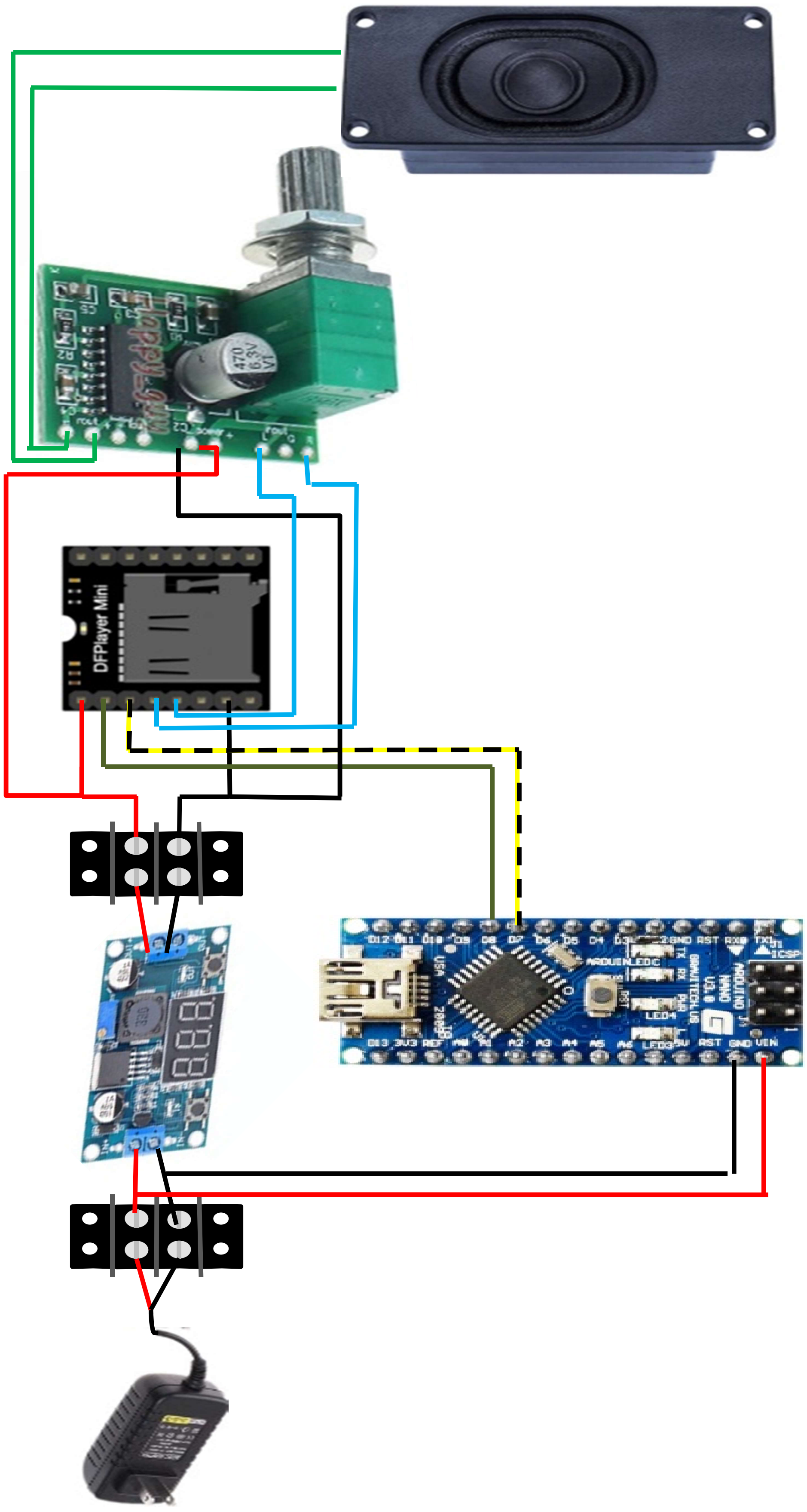
Now on to the wiring diagram. In this version I use a small amplifier between the MP3 player and the speakers, so I can change the volume without changing the software. Alternately, I could have eliminated the amplifier (and saved $4), but this would have required that I remove wires from pins 4 and 5 of the MP3 player, and then wire the speaker into pins 6 and 8 of the MP3 player. For my money, the amplifier is worth the extra cost. If you want to try it without the amplifier first, well it is a simple change later on to add an amplifier.

I have placed an Arduino sketch which plays the sound of a lumberjack felling a tree on GitHib at http://www.github.com/daackm, so visit the site and open the "Arduino-Animations-Tutorials", and copy the "Part 9" code. Open a new sketch in the IDE, delete all the original code, paste my code into the IDE and save the sketch to your PC.

I do need to emphasize the details about placing your files on the SD card. To play a sound file on the MP3 player I recommend, you need to reference each file by its numeric place on the card, not by its name; If you want to play the 3rd file, the line in the sketch would be "myDFPlayer.play(3);". Keeping track of which sound is in which file is tricky, and I suggest that you name your files starting with a number, such as "001Chopping Sound.mp3", or "002Falling Tree.mp3", and then copy the files to the SD card in order, one by one; if you want to update a sound file, you can't just replace it, you need to reload the entire set, one by one. And don't place more files on the card than you need for your current animation.

Also, take a look at the line "boolean play_state=digitalRead(8);". Notice that in the wiring diagrams we have a jumper running from pin 16 of the MP3 player to pin D8 of the Arduino. This "digitalRead" on Arduino pin 8 allows us to monitor whether the MP3 player is still playing a song, or has completed the task. In our demo we take a 10 second break after playing the sound file before we repeat the operation. If you look ahead at my Carousel project you will see in the code that I play multiple sound files with differing lengths, which I play in random order, so it is important to know when one song ends so I can start another tune at the appropriate time, and this is made possible by doing a digitalRead on pin 8 of the player.

I do owe a debt of thanks to Fernando Koyanagi and his video at https://www.instructables.com/id/MP3-Player-With-Arduino . His work was particularly helpful in getting my audio projects up and running.



4/28/2020 4:11 PM