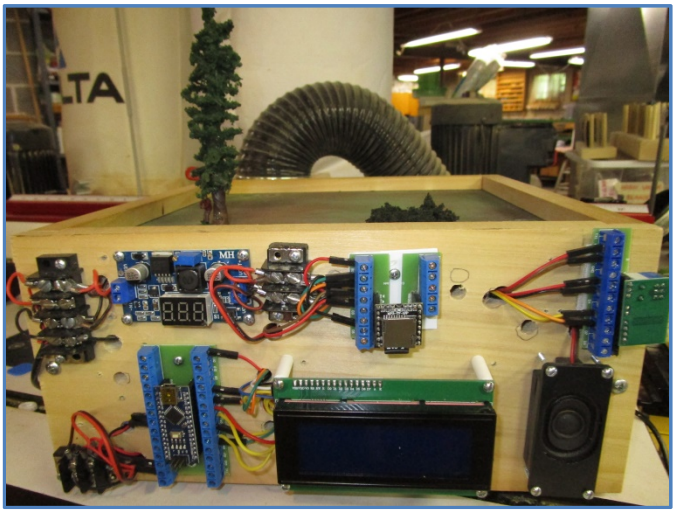
Project 4 - Lumberjack

July, 2020
David Ackmann – Gateway Division NMRA

Have you ever been to the Museum of Science and Industry in Chicago and lingered over “The Great Train Story” exhibit? Just past the port of Seattle, if you are walking clockwise, perhaps you saw the logging operation up on the hill. Did you push the button to make the tree fall? I did, more than once, and had to have a similar animation for my railroad. Here’s how I did it.

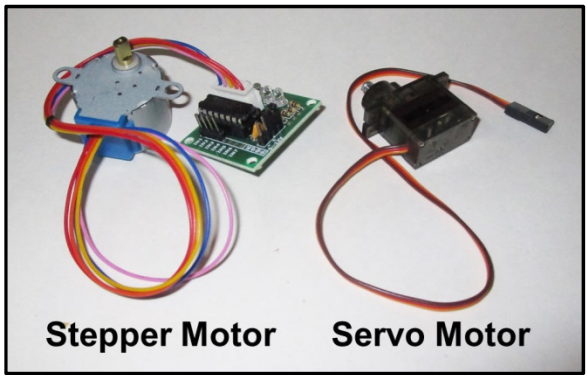


I can’t remember whether their animation had a lumberjack or just the falling tree, but I also wanted a lumberjack swinging an ax. My layout is pretty flat, so no hillside covered in pine trees for me. But more, my animation had to have sound effects and a fire to burn the unwanted branches. All would be controlled with an Arduino microprocessor.

Before we get any further, you can find the Arduino sketch that I used for this project at http://daackm.github.io, in the “Projects” section under the “Lumberjack” subheading. If you want to see it in action, visit http://daackm.github.io/Trailers.html (a case sensitive URL).

But where to start? For me, I decided where I wanted the tree to be placed, and where I wanted it to fall. I cut a slot in the layout surface, about 2.5” long and 0.75” wide; the tree would be positioned at one end. I wanted this to look as good as possible, so I bought a pair of Woodland Scenics’ “Premium Pine Trees”, and one of these would be the centerpiece, mounted on the arm of a small servo motor and rotated on demand, about 70° in all.

You might ask, “Dave, what motors did you use use”. I ended up using one metal geared motor from AMAZON (ASIN: B07NV476P7, 2 for $10) and one plastic geared servo motor (ASIN: B07MLR1498, 4 for $10). I wanted to use a servo instead of a stepper motor, because servos use absolute positioning, which means they always can be returned to a specific position on demand; not so with a stepper motor. I was concerned about using plastic gears because of their inclination to strip, but the metal geared servo was so noisy that I went with a plastic geared motor for the lumberjack figure.



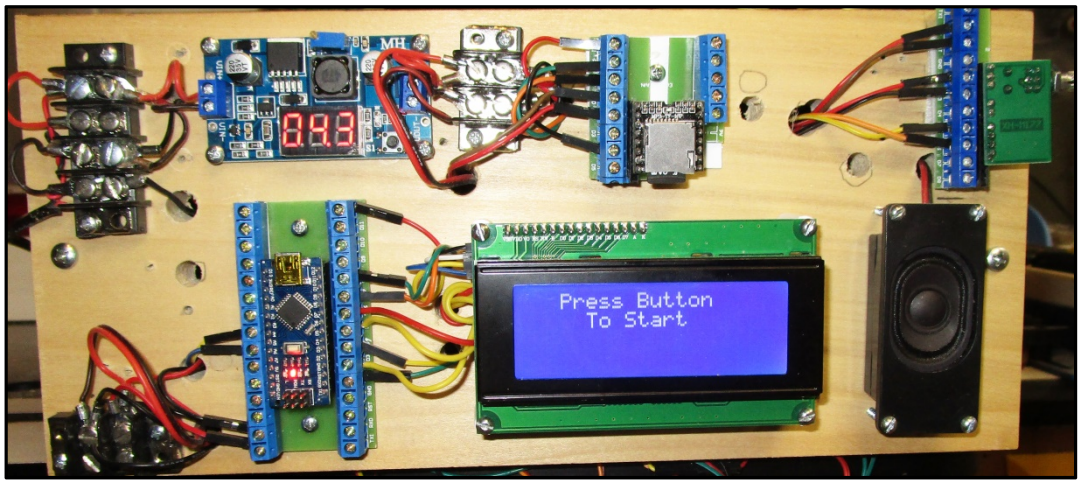
The tree was not mounted directly on the motor, but on an “arm” (aka: “horn”) which would be pressed onto a motor shaft, and this turned out to be a frustrating process. I tried to drill a hole through the trunk, but soon learned that the armature was composed of fine wires which got tangled during drilling, and once exposed did not present a good surface for attaching to the motor arm. What finally worked was a 6 step process:

1) I removed the pin from the bottom of the trunk;
2) I carefully sanded about a half inch up one side of the bottom of the trunk. I used a stationary belt sander which allowed me to use both hands to control the tree during the operation;
3) I epoxied the tree to the arm, making sure not to obstruct the mounting hole of the arm; I made sure that the round part of the arm which will mate with the motor shaft is pointing away from the tree;
4) I discovered epoxy was not enough to hold the tree to the arm, so I tightly wrapped the epoxied tree trunk in three turns of white Teflon tape usually used for sealing plumbing threads;
5) I covered the tape with another layer of epoxy;
6) I painted over the taped area with a mixture of brown and black acrylic paint.

To mount the tree motor to the layout, I needed a mounting block. I started with a piece of 1 by 2 lumber, about 2” long. I cut a notch in one side, near one end, just large enough to hold the servo motor. I drilled a 1/4" hole in one side to allow the wires to pass through, and then attached the motor to the block with the screws that came with the motor. I then glued the block into place (I like “Titebond” glue), near the edge of the slot, such that the motor shaft extends into the slot about 1/4". More than once I had to remove the block from the layout, and a moderate blow from a ball-peen hammer usually knocked it off in one blow; trying to remove it with a chisel was not as effective.

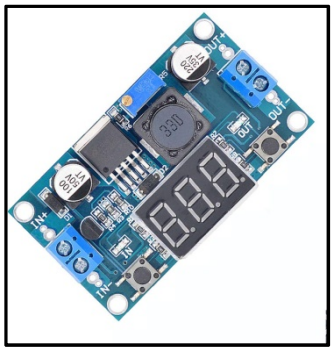
After the epoxy dried (I allowed it to dry overnight) it is time to mount the tree onto the motor. Press the round part of the arm onto the motor shaft and secure using the screw provided with the motor. Once you are satisfied with the position, connect the yellow wire from the motor to pin D4 of the Arduino using a male-male jumper wire.

Next comes the Arduino. I am going to assume that you have already read my documentation on why I like the Arduino Nano (its expansion board), how to power it with an external power supply, how to use motors, displays and how to add sound to a project, so I won’t go into details on these topics again; if you have not done so, visit https://daackm.github.io and start at the beginning. You are free to use any wires you like, but I am pretty strict about using red or white wires for positive voltages, black for ground wires, and just about any other colors you like for data wires. I like 24 gauge wires for this project, and often use jumper wires directly plugged into expansion boards, then screwed down for a tight connection.



The photo above shows the overall wiring. From the upper left and going across the top row, unseen is a 12V power supply (aka: “Wall Wart”) wired into a barrier strip. Next is a “Buck Converter” (aka “voltage regulator”) which drops the voltage down from 12V to 4.3V as required by the components to the right, the MP3 Player and amplifier (an additional barrier strip separates these audio components from the Buck Converter). On the bottom row we have a third barrier strip which received its wires from the barrier strip above, and then passes that power along to the Arduino Nano and its expansion board. To the right of the Arduino is a four line Display, and finally a speaker.

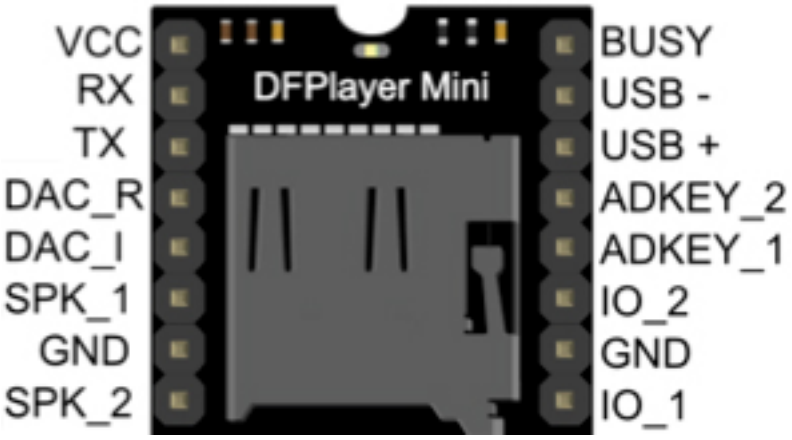
I started at the upper left and wired the power supply into a barrier strip. I knew I was going to need several points of 12V power, as well as several ground points, so I placed a jumper wires across two points of the barrier strip for both 12V and Ground. My choice for a power supply is a 12VDC, 2 Amp unit, AMAZON ASIN: B07VBR327W, but these specific units tend to come and go, so you may need to look up a suitable unit by voltage and current instead of the AMAZON ASIN. Using a volt meter, I made ABSOLUTELY CERTAIN that I knew which wire of the power supply was positive voltage and which was the ground, and then wired them into the red and black positions of the barrier strip respectively (electrical components are not pleased if you supply power to them with the wires reversed). From the upper barrier strip, I ran red and black wires into the lower barrier strip, then to the VIN and GND points of the expansion board. Finally, I installed the Nano into the expansion board.



Just to the right of the upper-left barrier strip is a thing called a “Buck Converter”. Mount it to the fascia with small screws, and wire it in to the power supply’s barrier strip with black and red wires. But these things are a bit tricky, so before you hook anything up to the barrier strip downstream of the Buck Converter, dial down the Buck Converter using the blue trim potentiometer all the way counter-clockwise, until it reads about 1 volt; ***failure to do this first will fry the audio components.*** Part 9 of the web site at http://daackm.githug.io describes this task in detail.

The barrier strip downstream of the first Buck Converter also supplies power to the two motors, so when I later install the motors onto the layout, I run wires from that barrier strip to the black and red wires of the servo motors (the remaining yellow wire on the servo is used to control the motor’s rotation).

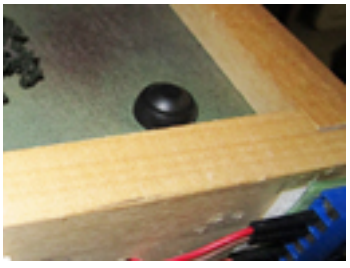
Make sure you understand Chapter 9 of the website at http://daackm.github.io before attempting to install the audio components, including the sections on making expansion boards for the MP3 player and amplifier, and I made one for both the MP3 player and the amplifier. From the first buck converter, I ran red and black wires to the MP3 Player's VCC (voltage input) pin and Ground pins (GND), to give it power. I ran signal wires from the Arduino to the MP3 Player; pin 7 from the Arduino to pin 3 of the MP3 player, and pin 8 of the Arduino to pin 4 of the MP3 Player. No wire from the "Busy" pin of the MP3 player to the Arduino is required for this project.



Next, I wired in the amplifier. I don't really like this amplifier because it does not come with pre-soldered header pins, so I have to solder them in myself, which is a tedious and delicate task. But solder the pins we must, long end of the headers away from the potentiometer, and so I did. I then placed the amplifier into its home-made expansion board. From the 4.2v buck converter's barrier strip, I ran a red wire from the powered connection point to the "Power +" pin of the amplifier, and a black wire from the ground position of the barrier strip to "Power –" pin of the amplifier, to give it power. To connect the MP3 player to the amplifier, I run any colored wires from pins 4 and 5 of the MP3 player to the "Input L" and "Input G" pins of the amplifier (signal and ground). Finally I ran wires from the amplifier's "L Out +" and "L Out –" pins to the speaker.

You can save $4 by eliminating the amplifier, removing the wires on pins 4 and 5 of the MP3 player, and then wiring the speaker into pins 6 and 8 of the MP3 player. For me, the amp and its volume control is worth $4, but the choice is up to you.

The animation is initiated by a momentary, normally open push button switch. I used one that comes in an 8-pack (AMAZON ASIN: B07XG4N6J1). Wire one side to pin D12, and the other side to any ground point, making sure to solder in a 10KΩ resistor somewhere in-line; cover the joints with heat-shrink tubing.



Next, I recommend that you install a display. This is covered in detail in Part 11 of the website (https://daackm.github.io). Installation is really nothing more than connecting the Arduino's A4 pin to the Display's SDA pin (male-female jumpers work well for this), the Arduino's A5 pin to the Display's SCL pin, and finally 5V and GND from the Arduino (or a barrier strip, the 4.2VDC will work fine) to the display (assuming that you bought the proper display, AMAZON ASIN: B071FGZX8G). The display tells a visitor to "Press Button To Start", and informs them to wait while the tree is on the ground and while the lumberjack is resting. You may have to turn the contrast up or down using the blue "trim potentiometer" on the "piggyback board" on the back of the display. Also, my display came with a hardware address of "0X27", but some may come with an address of "0X3F". If the display has power when the animation is plugged it, but doesn't display any messages, change the address around line 58 accordingly. The display is not absolutely essential, but it is good human engineering.

With all the components installed, power it up and rotate the blue potentiometer on the Buck Converter to its proper operating voltage of 4.2 to 4.3 volts.

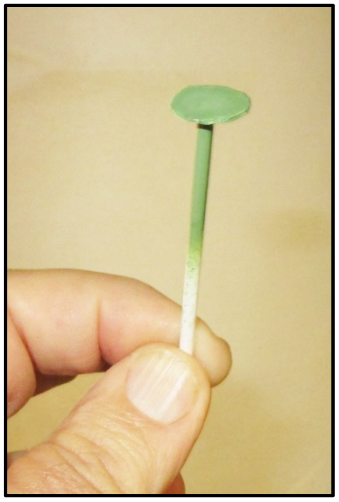
You are now ready to load the software into the Arduino. I will assume you know how to do this, so just open https://daackm.github.io and scroll down to the Lumberjack project. In there you will find an entry for the project's Arduino code, so click on it and the sketch will open in a "text-like" window. Highlight the entire file (a Control/A works well for this) and then use a Control/C to copy it to the Windows clipboard. Open the Arduino IDE and under the FILE tab, create a NEW sketch. Highlight and delete all the code which normally comes with a new sketch, then paste the Lumberjack code in its place (Control/V works for this, or you can click on the EDIT tab, then the Paste command). Save the sketch in a location of your choice.

Download the "Single_Chop.MP3" file to your SD card, and secondly (not at the same time as the previous file, these ***MUST*** be loaded separately), download the "Falling_Tree.mp3" file onto the SD card, and place the SD card into the MP3 Player.

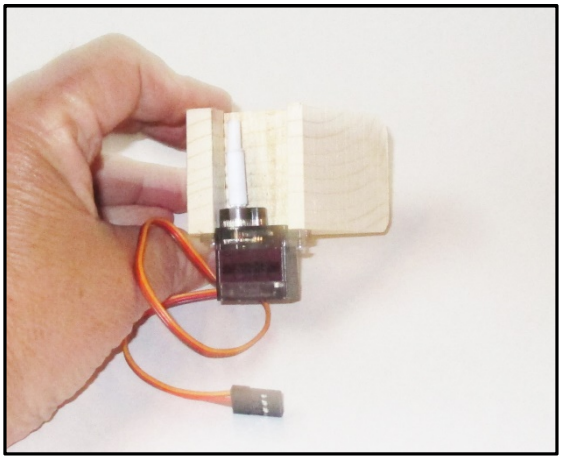
Press the start button, and you should hear the lumberjack's ax hitting the tree five times, and then hear and watch the tree fall (and even bounce once, kinda cool). The sketch waits 15 seconds before restoring the tree to its upright position, then 15 seconds more to allow the lumberjack time to rest. If you hear nothing, remember the first 3 laws of Computer Science are 1) check the cables; 2) check the cables again; 3)

have someone else check the cables. The project is designed to stall if it cannot successfully initialize the MP3 player, and failure to initialize is often caused by incorrect wiring or a missing DC card. It really helped me to build a spreadsheet of pinouts between the various components, and I encourage you to do the same.

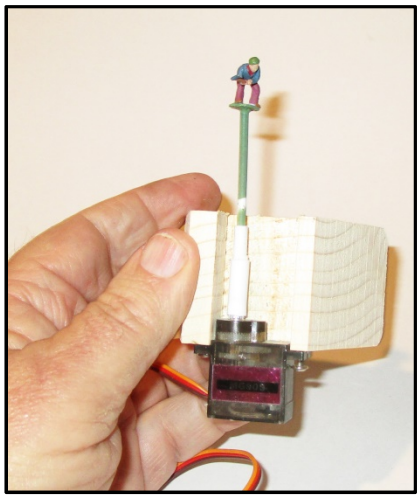
Now it's time to add the lumberjack figure, but how do we know where to place him on the layout? I temporarily placed my figure on the layout such that his ax was touching the tree, then moved his feet to a position I liked, and marked the spot between his feet. I drilled a #30 hole through the layout surface, and this would be the spot for an axle around which he would rotate.



Now the Lumberjack needs a platform on which to stand, and an axle on which to rotate. I made one by cutting out a 1/2" diameter circle of 0.03" styrene. I also cut a 1" piece of 0.010" plastic rod (Evergreen Plastics #213), glued it to the platform and painted it to match the layout surface. Once the paint dried, I inserted the rod through the hole in the layout. Next, I constructed a trapezoidal wooden block from a 2" x 2" piece of lumber about 3" long, and cut a notch through the short side, just large enough to hold the motor. I attached the motor to the bottom of the block with the small screws provided with the motor.



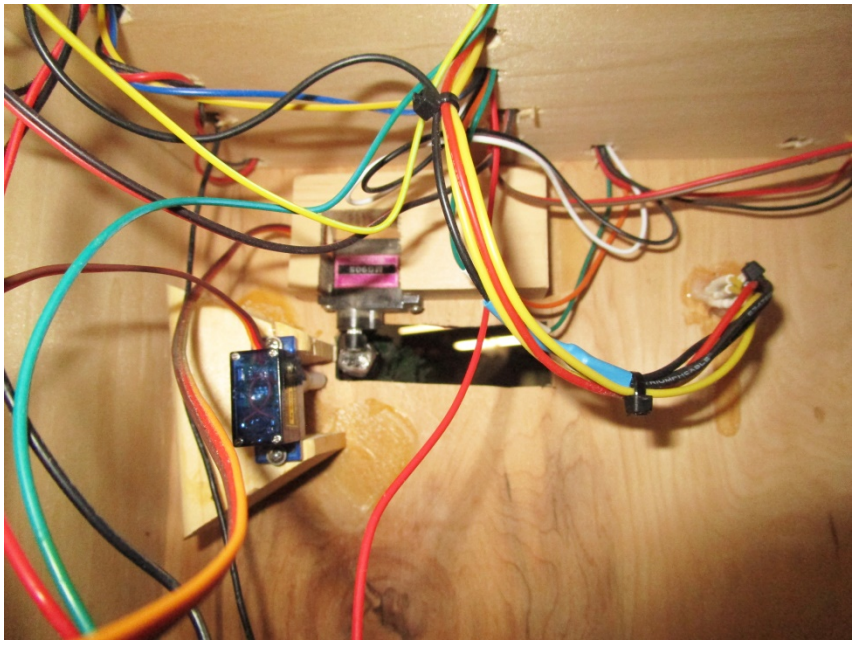
Now I needed a "transmission" to connect the motor to the lumberjack, and some way to align the mounting block such that the motor would align with the lumberjack's axle. I discovered that Evergreen Plastic #227 could be jammed securely around the drive shaft of the lumberjack's servo, but its inside diameter was too big to hold the axle. But if I inserted a piece of Evergreen #225 inside the #227 and glued it with Plastic Weld, it would stay secure, although the interior diameter was now a bit too small for the axle. I drilled out the transmission about 3/4" with a #38 bit, and the axle held snuggly, trimmed the end of the transmission so that it did not extend past the top of the mounting block.



I decided to enlarge the hole in the layout just a tad, using a #28 bit. I then inserted the axle into the hole (no glue on the axle yet), applied Titebond glue to the mounting block, aligned the transmission with the axle

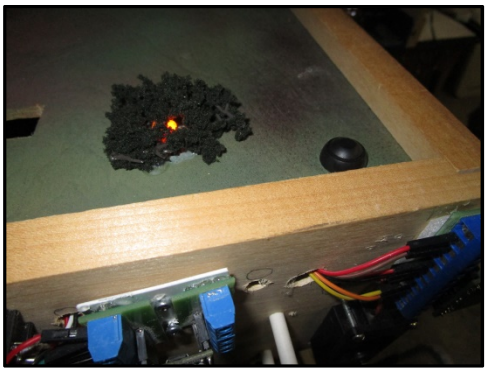
and glued the block into place, making certain that the “transmission” was straight, and that the block position would not obstruct access to the screw holding the tree to its motor. I let the block dry overnight. The next morning I extracted the lumberjack’s axle and platform and cut it down so that the platform was barely above the level of the layout (a piece of wax paper makes a good spacer). I then gave the rod a coat of Plastic Weld glue and inserted it into the transmission, making sure the rod actually was inserted into the transmission cylinder (it is VERY EASY for the cylinder to wander out of position), and let it dry for several hours.

The picture to the right shows the final positioning of the mounting blocks for the tree and the lumberjack. Note how the tapered ends of the lumberjack’s mounting block are positioned such as to not obscure access to the screw holding the tree to its motor. It took me several times to get that mounting block positioned just right.



With the tree and lumberjack’s motors installed, it is time to wire their motors to the Arduino. The servos have female connector at the end of their wires, so some male-male jumpers would work well to connect them to the Arduino; the lumberjack’s wire goes to Arduino pin D2, and the tree’s wire to pin D4.

Once the lumberjack mounting block, transmission and platform rod were secure, I needed a way to position the platform so that the Lumberjack’s ax was touching the tree when his motor had rotated into the “chop position”. To do this, I pushed the button to start the animation, and immediately upon hearing the first “Chop”, I cut the power. I applied some clear-drying tacky glue to the platform and positioned the lumberjack figure so that his ax was touching the tree and his feet were in the center of the platform, then let the joint dry overnight. At first, the figure tended to fall until the glue was about 15 minutes into its cure, so keep an eye on it for a while, or perhaps wait 10 minutes or so before moving the lumberjack into place. The next morning you might want to put a second coat of glue on the lumberjack’s feet, as this joint is rather fragile.



It's not absolutely essential, but I wanted a pile to burn some brush. We covered a Dumpster fire in Project 1, and a Brush fire is the same thing, except that the LEDs are covered with limb cutoffs from a pine tree. But if you want to see how another modeler made this work, check out a YouTube video from "Ron's Trains and Things" (https://www.youtube.com/watch?v=wBn7pHEldWI). His video is also a great introduction to Arduinos, and at the end (about 14 minutes into the video), Ron describes how to use LEDs to make a campfire, and I liked is presentation. I built one as a pile to burn discarded brush from the logging operation, using Pulse Width Modulation on digital pins 3, 5 and 6. Normally pins 9,10, and 11 are also available for use as PWM pins, but if a servo motor is also in use, as is the case with this project, pins 9 and 10 lose their PWM ability.

That just about completes this project. Sure, you might want to add a few more trees and shrubbery, maybe a few rocky outcroppings, and maybe a logging truck or a few stumps, but you get the general idea. If I made another one of these, I would go with a higher quality servo, specifically one where I could get aftermarket horns (not available on the lower cost 9g servos); the customer support provided for the servos I used was uncommonly poor. Instead of a single-arm horn, I would use one with an additional arm at a right to the tree, and on the additional horn I would attach a wooden rectangular plug to cover the hole, so that the hole wouldn't be so obvious when the tree was standing, and the right angle horn would allow the plug would rotate out of the way when the tree was falling. I also might animate a chainsaw figure cutting branches, which would require a second MP3 player. I still have pin 11 available and could make the fire even bigger. Maybe I should add a relay to control a smoke generator. And for a comic touch, I might add a relay and motor to animate a bulldozer to push the tree back to the upright position.

For me, it's been a fun and challenging project. I hope you enjoyed reviewing it.