FOUR BANGER

Four Banger v1.2
Welcome to the official online manual for the Four Banger project! This manual is a work in progress, so please bear with us as we try to bang this thing out. Rather than place a copy of this manual in the \docs folder of the official download package, the plan is to keep the "master" online, so that users may always link to the latest and greatest content.

What is Four Banger?

If you are reading this manual, chances are you already know what this project is- but just in case, here is the lowdown:

A four channel prop controller with MP3 audio for about $20

This project will show how to build a 4 channel prop controller with MP3 quality audio for about $20. It will also take about 15 minutes to assemble by anyone with basic soldering skills. The goal of this project is an affordable controller that is easy to use, but versatile enough to power a professional-grade animatronic prop. The controller is designed to be fully functional without the need for any external components (resistors, transistors, etc...)

Goals:

- **Affordable**
- **Easy to Assemble**
- **Compact Size**
- **Simple Enough for a Beginner**
- **Configurable Enough for Professional Use**

This controller will use an Arduino Mini Pro with the Atmel AT328p processor. Configuration options and sequence data will be stored using the AT328p's onboard 1k eeprom. The controller will be programmable using either manual a button board (button banging), or via a custom PC-based programming application.

Shopping List (note that these prices include shipping):

- **Arduino Nano 5v AT328p** **$2.41**
- **4 Relay Module** **$3.20**
- **PIR Sensor** **$1.11**
- **Serial MP3 Module by Catalex** **$7.37**
- **1gb SD Micro Card** **$3.19**
- **12vdc 500ma Power Adapter** **$2.18**

TOTAL---------- About 20 Bucks

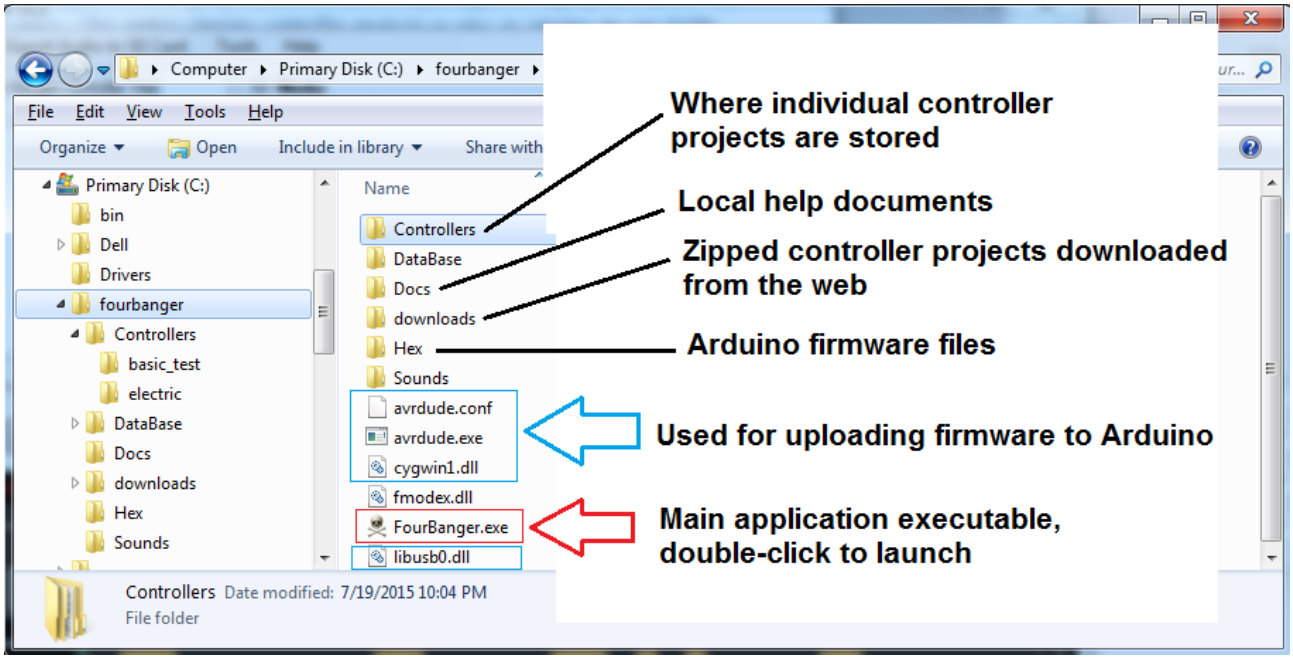
Key Concepts and Terms

Many of the same terms will be used over and over in this document, so a good place to start will be to define the more common ones.

- **Controller** The term Controller may be considered a "project". In the Four Banger program, all of the custom configurations and settings that are made to change the way the prop controller behaves are saved into a single file for each project. Each controller project is also allocated its own subfolder. When audio files are associated to each controller project, they are copied into that project's subfolder so eveything for that project may be found in one place. This makes sharing controller projects as easy as zipping up one folder and sharing it with your friends*
- **Scare Sequence** Each controller has one scare sequence. The scare sequence, or just "the sequence", is the object that keeps track of what happens when the controller is triggered- like what audio file to play and how long to play it, etc..
- **Trigger** This is the action that causes the controller to begin playback of the scare sequence.

Application Folder Structure

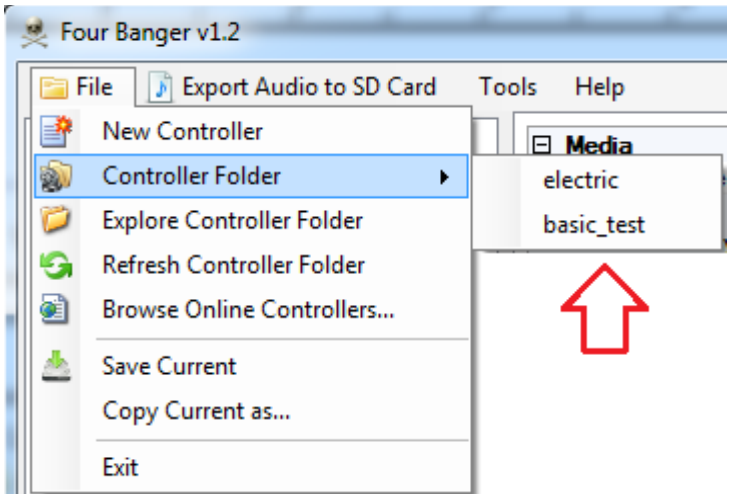
The Four Banger application is currently distributed as a single zip file containing the following directory structure. I typically install it right on the root of the c:\ drive on my PC, but you can really place it anywhere you like- just remember where you put it!



Program Startup

When the Four Banger application loads, the first thing it does is load up the contents of the \Controllers subfolder into the file menu. The \Controllers subfolder is the location where all of your controller projects are saved to physical files on the PC hard drive.

Above you can see two project subfolders. Each project subfolder will contain a file with the naming convention of *.fbc and usually two mp3 files (the ambient and scare tracks). The *.fbc file is the project file.

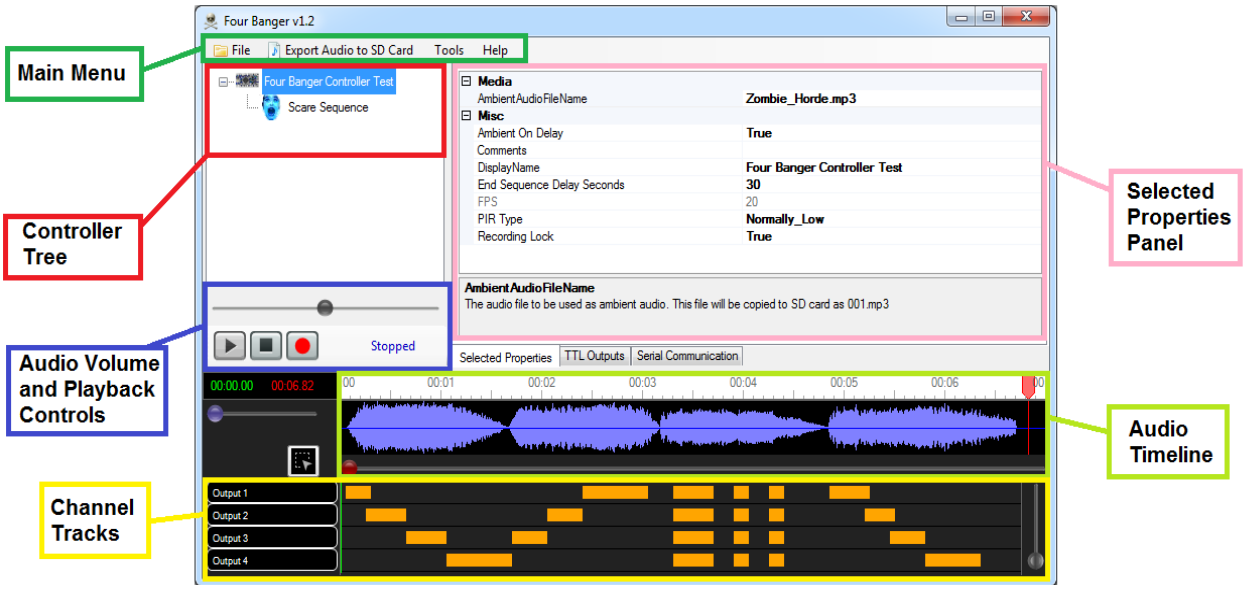


You can see that from the File menu, the Controller Folder menu contains selections that match the directory structure found on the hard drive. To open one of the controller projects, just select the one you want to open from this menu.

If for some reason the \Controllers folder is completely empty when the application is started for the first time, you will be immediately prompted to create a new controller project. Simply type in some name for the new controller and click Save. Since the app needs a controller project to be loaded at all times, failure to create at least one controller will cause the app to close.

Software Interface

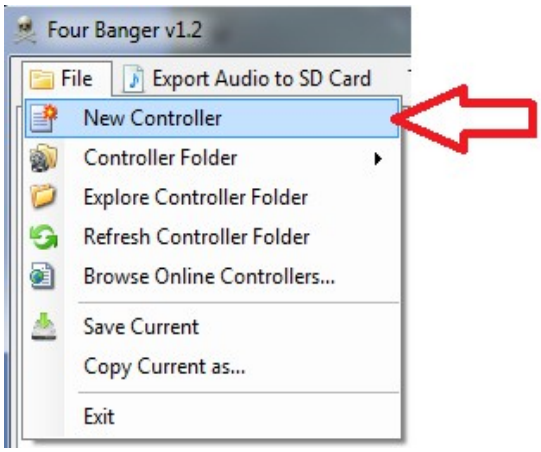
The Four Banger main window is comprised of the following main areas:



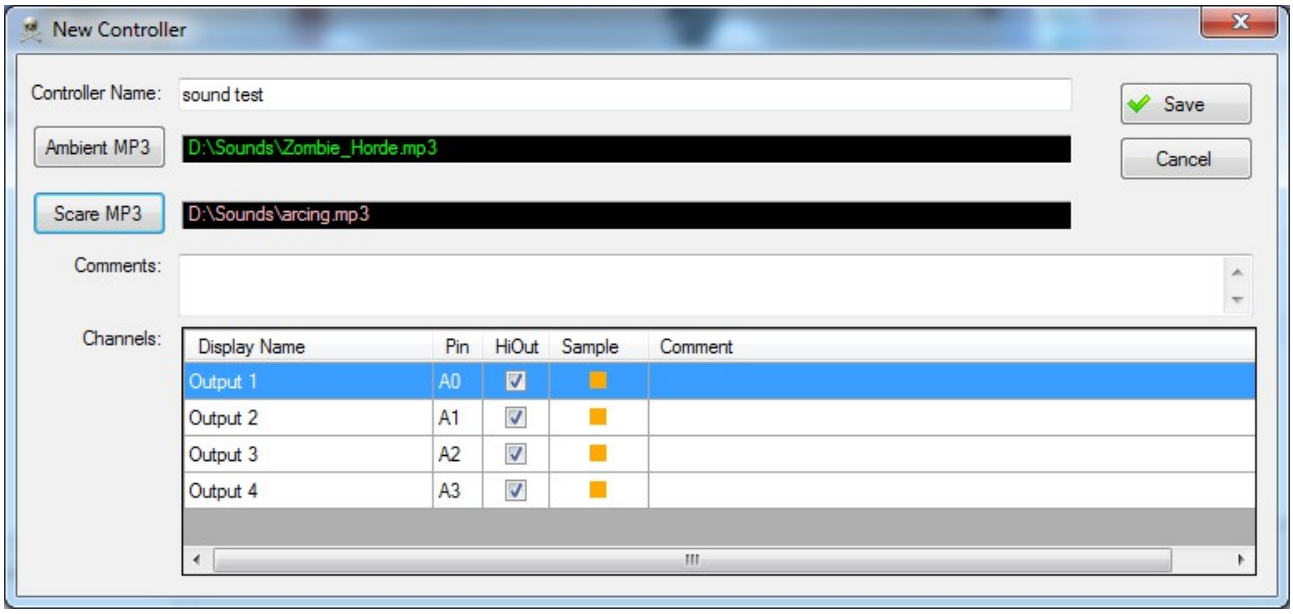
- **Main Menu** This is a typical Windows menu bar used for opening & saving controller files, accessing help, etc... Other functions called from this menu include copying audio files to a micro SD card and uploading firmware to your Arduino
- **Controller Tree** Clicking on the controller node in this tree will cause the controller's properties to be displayed in the Selected Properties Panel. Clicking the scare sequence node will cause the scare sequence's properties to be displayed in the Selected Properties Panel
- **Audio Volume and Playback Controls** Adjust volume, start, stop, and record buttons
- **Channel Tracks** This is where the On/Off states of the 4 TTL outputs are displayed and edited
- **Selected Properties Panel** This panel displays and allows editing of both controller and sequence properies such as audio files, sequence duration, PIR type, etc. Most properties have additional help text that is displayed when the property is selected in the grid
- **Audio Timeline** This is where the audio waveform is drawn and a ruler & carat display the current point in time
- **TTL Outputs** This tab allows viewing / editing the properties of each of the four TTL output channels. Properties of each channel include Display Name, Color, Default Hi/Low, etc..
- **Serial Communication** This tab is where you connect to the Arduino. It contains a message window to display communications sent back from a connected Arduino. Uploading sequences and configurations take place here

Creating a New Controller

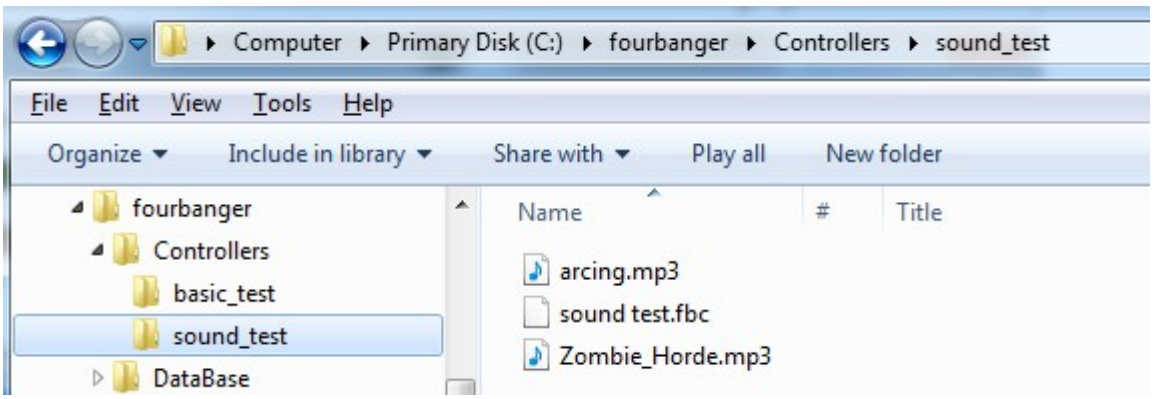
To create a new controller project, choose File, New Controller from the main menu



This will pop up the new controller dialog window. Here you can enter a name for the new project as well as the ambient and scare audio files that are to be associated with the project.

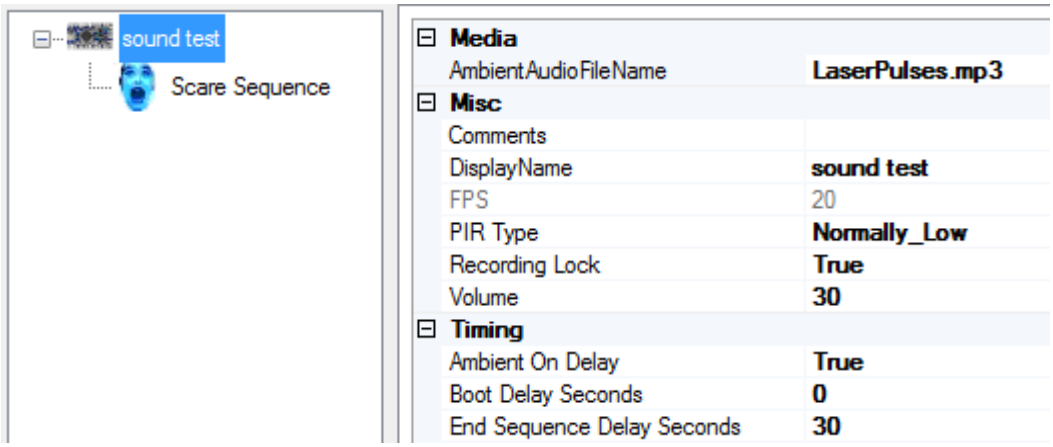


When a new controller is created, the application will create a new folder for the project:

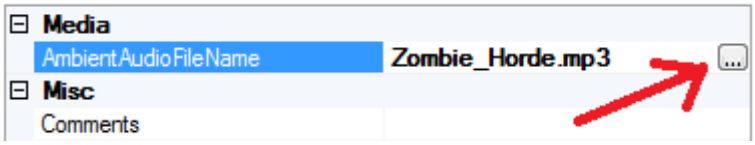


Controller Properties

The main properties of the controller may be viewed/modified by selecting the controller in the controller tree



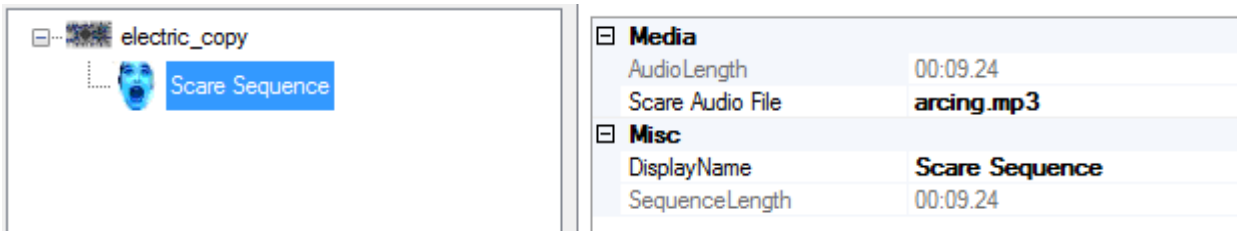
- **Ambient Audio FileName** this is the audio that plays in the background while the controller is waiting to be triggered. To select a different audio file after the controller project has been created, select the property in the grid/panel, and then click the [...] browse button that appears to the right



- **Comments** this is just a scratchpad where notes about the project may be entered
- **Display Name** a friendly name for your project, upon creation this will be the name of the subfolder where the project files are located
- **FPS** frames per second. This determines resolution & max sequence duration
- **PIR Type** most are Normally High, but you can change to Normally Low if needed
- **Recording Lock** locks out the manual button board's recording button
- **Volume** adjust the audio level of the MP3 module (1 = quiet, 30 = loudest)
- **Ambient On Delay** true means to go back to ambient audio before the **end sequence delay**. A setting of false means that the scare audio will continue to play (or end and go silent depending on the length of the track) during the **end sequence delay**.
- **Boot Delay Seconds** this is an optional delay (0 to 255 seconds) that the controller will wait after booting up. This is just to help avoid immediate triggering of props when they are turned on.
- **End Sequence Delay Seconds** this is an optional delay (0 to 255 seconds) that the controller will wait after a scare sequence is played. The controller is not triggerable during this wait period.

Scare Sequence

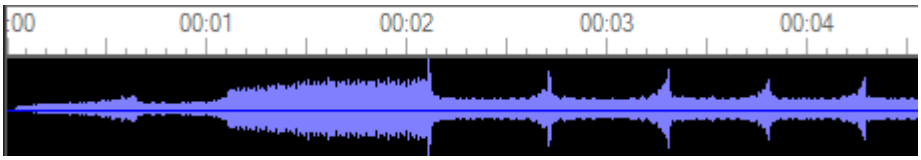
Clicking on the Scare Sequence node of the controller tree will display the sequence properties.



The primary property of concern here is the Scare Audio File. If a scare audio file was not selected when the controller project was created, or a different file is desired, select the "Scare Audio File" property then click the [...] browse button that appears to the right.

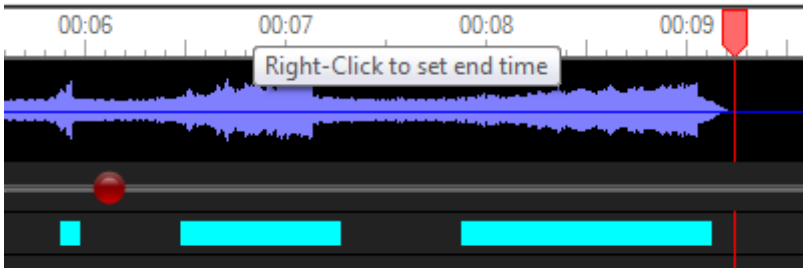
When a new audio track is selected, a few things happen behind the scenes automatically:

- A copy of the audio file is made into the project folder. This keeps the project files all together
- The audio file is sampled in order to generate an audio wave like this:



The sampling takes a few seconds, so if you see a delay after choosing a new file, that is what is causing the brief delay. These audio samples are also saved to the hard drive as part of the project, so the sampling only needs to happen once. This allows saved sequences to load up much faster.

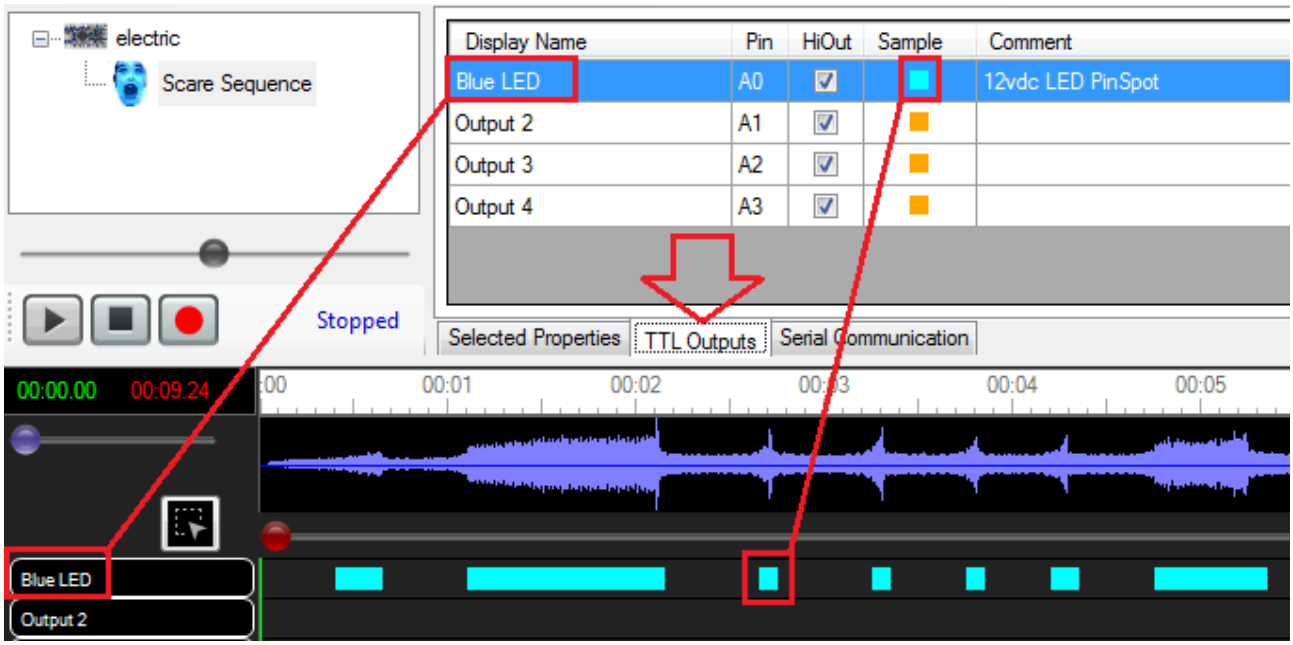
- The sequence duration is set to the same length as the selected audio file. If a different duration is desired, you may select a different end point by right clicking on the ruler portion of the timeline.



Since the scare audio track is only played once and does not loop, a scare sequence that is longer than the audio track will merely have no sound at its end.

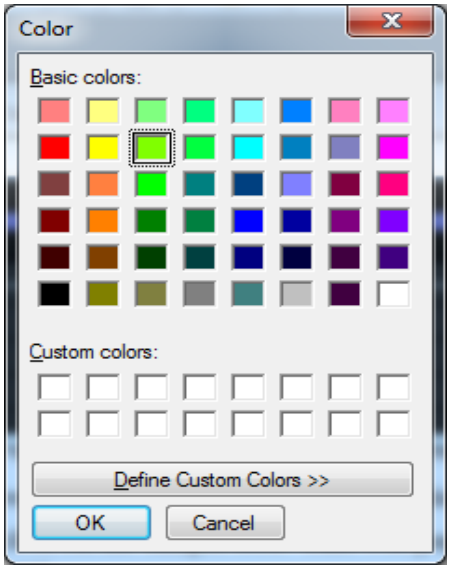
Channel Properties

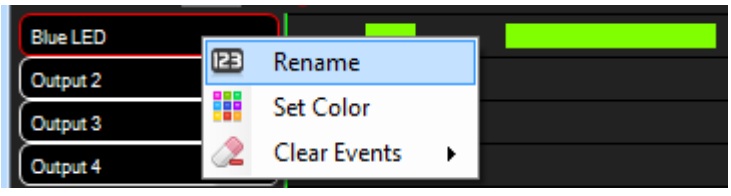
The TTL Outputs tab allows you to view/customize the properties of each of the outputs on the controller.



- **Display Name** changes the label that is displayed next to each track
- **Pin** this is a reminder of the physical output pin on the Arduino
- **HiOut** when true (checked), the TTL output of the channel will default to HIGH. When an event occurs, the TTL output will go LOW. Unchecking this box will invert the logic.
- **Sample** this is the color that the channel's events will be drawn in (on the track timeline)

  To change the selected color for a channel, click on the sample box. This will pop a color picker to select pretty much any color you want.





Name and color properties for each channel may also be changed by right-clicking on the channel's track itself.

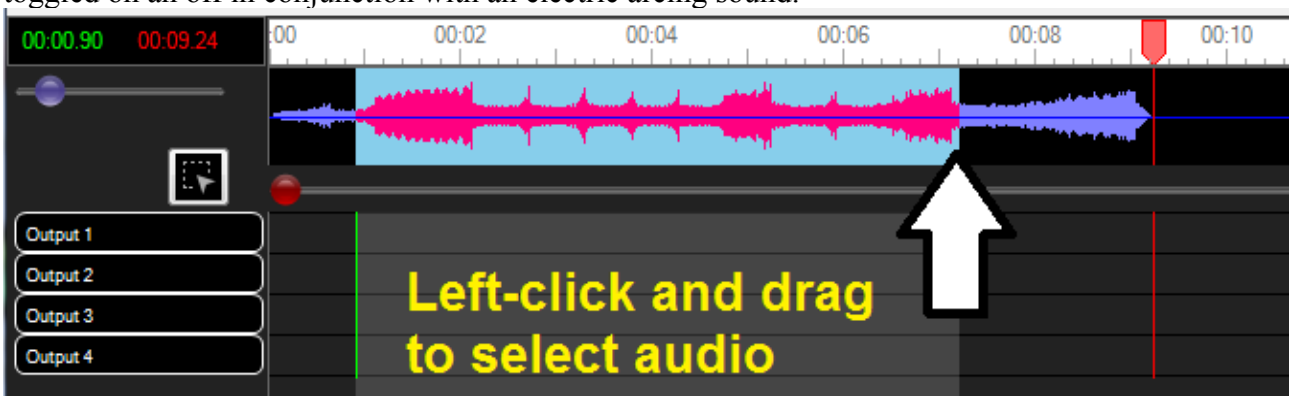
Serial Communication

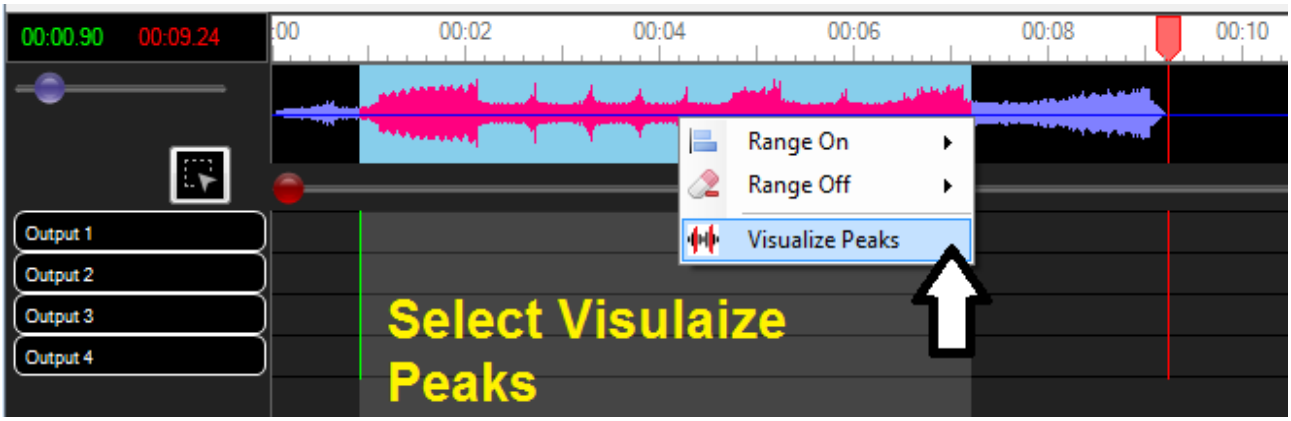
This tab is where the PC application talks to the Arduino.

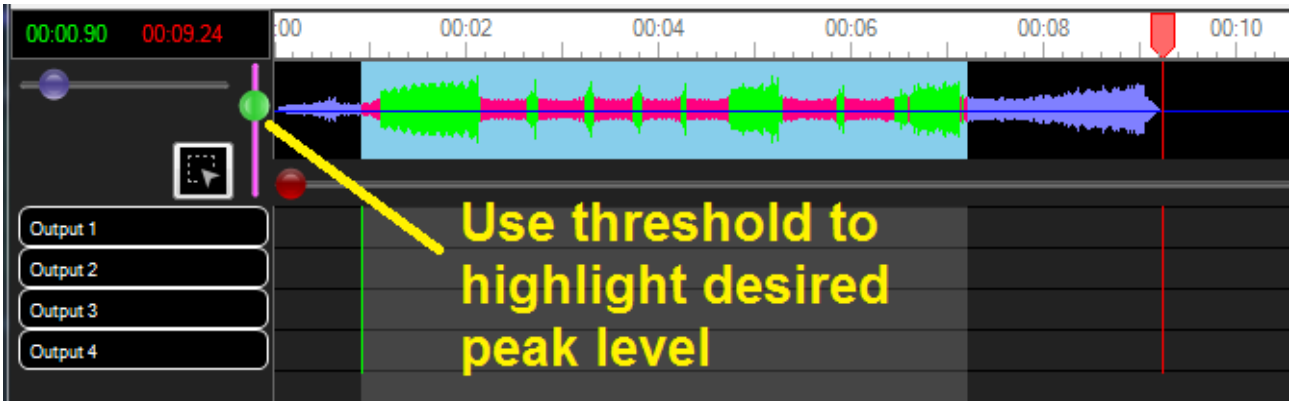
Audio Peaks

Occasionaly you may want your animation to occur in exact time with the scare audio. This may be done by selecting the audio in the timeline, then right-clicking to access a few special menu functions.

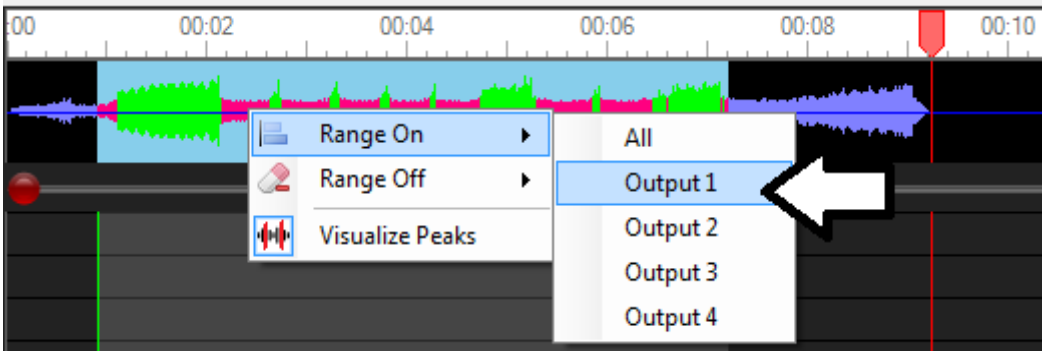
This technique was used in the electric arcing controller example project. It allows for a relay to be toggled on an off in conjunction with an electric arcing sound.



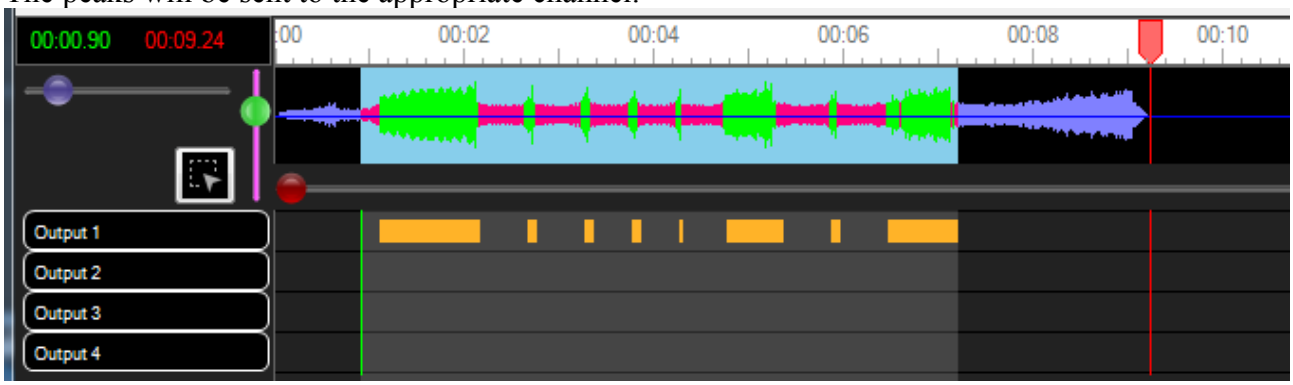




Right-click and select Range On, then select the desired channel to sent the peak data to.



The peaks will be sent to the appropriate channel:



Basic Controller Logial Flow