





LEDBlinky
Arcade LED Control Software
Hardware Support for LED-Wiz, Pac-LED64, Pac-Drive, and U-HID
Version 5.1
Created by Arzoo (2012)

Installation and Configuration Guide

Overview.....	3
LEDBlinky Website	4
MaLa, AtomicFE, and GameEx Plug-in Installation	5
HyperSpin, Maximus Arcade, and Stand-Alone Installation (for any Front-End)	6
How Do I Get This Thing To Work?	7
Configuring Other Emulators (not MAME)	8
It's Not Working, What Should I Do?	9
Running LEDBlinky in Stand-Alone (Command Line) Mode	10
LEDBlinky Core Application 	12
LEDBlinky Configuration Application 	13
Menus	13
Game Options	15
FE Options	20
MaLa / AtomicFE / GameEx / HyperSpin / Maximus Arcade Options	23
MAME Config	25
Audio Animation	27
Misc Options	34
Game Specific Animations	40
Generate LEDBlinky Input Map Application 	41
Menus	45
LEDBlinky Controls Editor 	46
Menus	47
Emulator and ROM/Game Lists	49
Edit Emulator.....	52
Edit ROM/Game and Controls.....	54
Edit Front-end (FE) and Controls	58
Edit Individual Control	60
How to Specify JDR Mode/Map Overrides	64
File Descriptions	65
Credits	68
Index.....	69

Overview

LEDBlinky is arcade LED control software for use with the GroovyGameGear LED-Wiz™ 32-port USB Lighting and Output Controller and the Ultimarc Pac-LED64™, Pac-Drive™ or U-HID™ USB Output Driver Boards. With the LEDBlinky plug-in, you can light your control panel's LEDs to indicate which controls (buttons, joysticks, trackball, etc.) are active during game play (MAME or other emulators), plus many other features. Here's a list of the current features:

- Supports the LED-Wiz, Pac-LED64, PAC-Drive, and U-HID USB output control devices. You can use multiple combinations of any of these devices.
- Supports MaLa, AtomicFE, and GameEx integration via plug-in OR stand-alone operation for HyperSpin, Maximus Arcade, or any arcade Front-End (FE) software.
- Light active and/or inactive controls for MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade user interface.
- Light active and/or inactive controls for MAME emulation.
- Light active and/or inactive controls for other emulators.
- Easily configure control attributes (color, intensity, spoken action, input codes, etc.) on a game by game basis, or define defaults for each emulator.
- Remapping controls (MaLa, AtomicFE, GameEx, HyperSpin, Maximus Arcade or MAME) will automatically remap the associated LEDs.
- With RGB LEDs, you can specify colors for individual controls or using a pre-defined colors.ini file, match the original game control panel button colors. Colors or intensities can also be customized on a game-by-game basis.
- Use audio output (music or game sounds) to blink, fade, or animate LEDs – great for use with Jukebox software.
- Blink and speak front-end UI controls by pressing a pre-defined “Help” button.
- Blink and speak controls when pausing a game and/or play a LED animation (selected, random, random montage, or specific to each game) or use audio output (music) to animate the LEDs. This is a MAME only feature.
- Flash start buttons when credits are available - this is a MAME only game dependent feature.
- Light start and coin buttons based on active player count for the current game.
- Flash all or active buttons when any is pressed.
- Full support for other MAME Outputs - light LEDs based on any output. Outputs can be linked to controls (P1_Button1, P2_Button2, etc.) or directly linked to a Device/Port.
- Extensive audio animation options let you completely customize how the LEDs blink to music or game sounds.
- When starting a game, LEDBlinky can play a LED animation (selected, random, or specific to each game), speak the game name, speak each button “action” while blinking the button in its correct color, speak the primary controls, and speak a custom message. When speaking the game name or custom message, LEDs can blink in sync with the speech.

- While playing a game, LEDBlinky can play a continuous LED animation (selected, random, random montage, or specific to each game) or use audio output (game sounds) to animate the LEDs. The LED animation will only effect unused controls.
- When starting or quitting the front-end, LEDBlinky can play a LED animation (selected or random), and speak a custom message. When speaking the custom message, LEDs can blink in sync with the speech. An LED animation can also play whenever the front-end UI is active (including audio animations).
- When changing game lists or emulator lists, LEDBlinky can play a LED animation (selected or random).
- When the Front-End and/or Screen Saver is active, LEDBlinky can play a continuous LED animation (selected, random, or random montage) or use audio output (music) to animate the LEDs. Random custom messages can be played at predefined intervals. Berzerk mode is also available for the random messages.
- Other speech features are available – Choose from multiple voices (downloadable) and set the voice rate and volume.
- Run (independent) LED animations for cabinet lights. These animations can run for all standard LEDBlinky events; FE Start, FE Quit, FE Active, FE Screensaver, Game Start, Game Pause, and Game Quit. Animations can also be specified for individual games. This feature requires an additional LED output control device for the non-control panel LEDs.
- From the front-end user interface, game controls can be lit as you scroll through the game lists. This feature is currently only supported by MaLa, AtomicFE, GameEx, HyperSpin and Maximus Arcade.
- Designate LEDs as “Always Active” for use with coin, start, or administration controls.
- Cocktail Mode lights all player controls for multi-player alternating games.
- Use the LED-Wiz built-in blinking effects.
- Set the GP-Wiz49 (for 49-Way joysticks) or Ultimarc UltraStik 360 joystick digital restriction based on the currently selected game’s primary control.
- Switch the Ultimarc ServoStik’s restrictor plate between 4-way and 8-way based on the currently selected game’s primary control.
- Support for pre-defined controller files (X-Arcade, SlikStik, etc).
- No Microsoft COM control dependencies.
- Use the LEDBlinky Animation Editor to create your own animations.

LEDBlinky Website

For software updates and online support, please visit the [LEDBlinky website](#). I have also documented the software development, along with my MAME project and other arcade related stuff on my [Arcade Addiction](#) site.

MaLa, AtomicFE, and GameEx Plug-in Installation

Unzip all the LEDBlinky files into the MaLa, AtomicFE, or GameEx *plugins* folder. Plugin specific files (*LEDBlinky.mplugin*, *LEDBlinky.atoplug*, *LEDBlinky.plugcfg*, *LEDBlinky_GX.dll*) must reside in the plugins folder. All other LEDBlinky support and configuration files must reside in the LEDBlinky folder below the plugins folder. Animation files (*.lwax or *.lwa) must reside in the lwa folder below the LEDBlinky folder.

```

...\Plugins
LEDBlinky.mplugin      (MaLa)
LEDBlinky.atoplug      (AtomicFE)
LEDBlinky.plugcfg      (AtomicFE)
LEDBlinky_GX.dll       (GameEx)
  ...\Plugins\LEDBlinky
  Animation Editor.pdf
  Bass.dll
  Color-RGB.ini
  Colors.ini
  GenLEDBlinkyInputMap.exe
  Install and Config.pdf
  LEDBlinky.exe
  LEDBlinkyConfig.exe
  LEDBlinkyControls.xml
  LEDBlinkyControlsEditor.exe
  LEDBlinkyAnimationEditor.ini
  LEDBlinkyAnimationEditor.exe
  LEDBlinkyTroubleshooter.exe
  Ledwiz.dll
  LibUSB0.dll
  Mame.dll
  MameOutputs.ini
  MameOutputTest.exe
  NewInputCodes.ini
  PacDrive.dll
  Settings.ini
  ZipDll.dll
  ... \Plugins\LEDBlinky\lwa
  <animation files *.lwax or *.lwa>
  ... \Plugins\LEDBlinky\jdr
  ... \Plugins\LEDBlinky\jdr\GPWiz49
  <files required for GPWiz49>
  ... \Plugins\LEDBlinky\jdr\UltraStik
  <files required for UltraStik>

```

The LEDBlinky plug-in should not be used in conjunction with any other MaLa, AtomicFE, or GameEx plug-in that controls the LED-Wiz, Pac-LED64, Pac-Drive, or U-

HID. If you have installed another LED plug-in, it should be disabled or removed from the *plugins* folder.


The LEDBlinky plug-in should not be used in conjunction with any native front-end feature that controls the LED-Wiz, Pac-LED64, Pac-Drive, or U-HID. Please disable any MaLa, AtomicFE, or GameEx LED features.

HyperSpin, Maximus Arcade, and Stand-Alone Installation (for any Front-End)

Unzip the LEDBlinky files. All files must remain in the LEDBlinky folder. The LEDBlinky folder may be located in any other folder. You may delete the *LEDBlinky.mplugin*, *LEDBlinky.atoplug*, *LEDBlinky.plugcfg*, and *LEDBlinky_GX.dll* files (only used by MaLa AtomicFE, and GameEx).

How Do I Get This Thing To Work?

Step 1

 Run the Generate LEDBlinky Input Map application (*GenLEDBlinkyInputMap.exe*). Before you can use LEDBlinky, you must create your input map.

The input map defines the relationship between each wired port on your LED device(s), and the keyboard input code for a button or input codes for other controls (Joysticks, Trackball, etc). It also allows you to assign a Port Label and LED Type to each wired port. The Port Labels serve two purposes, they tie together three ports for RGB LEDs, and they provide an easy reference to the ports from within the other LEDBlinky tools. The LED Type defines LEDs as Single, Red, Green, or Blue.


For example; let's say LED-Wiz ports 1, 2, and 3 are wired to an RGB LED under a button. That button is wired to the keyboard encoder and assigned to the letter "A". We also have a LED under our TrackBall wired to port 4. The input map would have four entries:

Port	Port Label	LED Type	Input Codes
1	P1B1	Red	KEYCODE_A
2	P1B1	Green	KEYCODE_A
3	P1B1	Blue	KEYCODE_A
4	TRACKBALL	Single	MOUSECODE_1_ANALOG_X MOUSECODE_1_ANALOG_Y MOUSECO...
5			

A full description on how to use this application is provided in a later [section](#) of this document.

Note: The input map is not necessary if LEDBlinky will only be used to set the joystick digital restriction (JDR) mode for Ultrastik 360 or 49-Way joysticks.

Step 2

 Run the LEDBlinky Configuration app (*LEDBlinkyConfig.exe*). This application allows you to configure the various features of the LEDBlinky plug-in and provides quick access to the other LEDBlinky tools.

Step 2a

From the "FE" menu, select your Front-End software (current options: MaLa, AtomicFE, GameEx, HyperSpin, Maximus Arcade, or Other).

Step 2b

For MAME emulation, you'll need to provide additional information – On the "MAME Config" tab, set the paths to the required (and optional) MAME files and folders. The colors.ini file is used to set game specific control colors. A version of this file with authentic control panel colors supporting over 1200 ROMs (at last count) has been

included with LEDBlinky. An updated version may be available for download from the ArcadeControls forum - Thanks headkaze!

Step 2c

From the "Game Options" tab, you can enable LEDBlinky to blink each button and speak the button "action". You can also configure LEDBlinky to play a LED animation, speak the game name, and speak a custom message, all before the game starts. Other Game options are available.

Step 2d

From the "FE Options" tab, you can configure various "Attract" features when the front-end starts, when switching lists, when the screen saver is active, and when the front-end quits. MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade specific features are also available.

Note: Features are only available for front-end's that support these events (Start, Screensaver, Quit); see [Stand-Alone Mode](#).

A full description on how to use this application is provided in a later [section](#) of this document.

Configuring Other Emulators (not MAME)

This is a manual process ☹. Since there is no way for LEDBlinky to know the control-input mapping (button assignments), or specific controls, or button colors for each emulator or individual game, you must provide the information manually using the LEDBlinky Controls Editor.

LEDBlinky and the Controls Editor provide a number of features to ease the configuration of emulators and games. Each time you play an unknown game (one for which no unique controls are defined), the Emulator and ROM/Game name is stored. From the Controls Editor import menu, you can display the list of unknown games and select which you wish to import. You can then define the controls. Controls for player 1 can be copied to players 2, 3, and 4. If the controls for one game are similar to another, you can copy the entire ROM/Game.

It is not necessary to define the controls for every game – using the Controls Editor you can define a default set of controls for each emulator (or even a default for all emulators).

A full description on how to use the LEDBlinky Controls Editor application is provided in a later [section](#) of this document.

It's Not Working, What Should I Do?

If LEDBlinky is missing critical information such as a required file or configuration value, a message will be displayed after you exit the application and provide you with the option to display the log file. LEDBlinky logs all errors to the *LEDBlinky.log* file in the *LEDBlinky* folder. These types of problems can be easily resolved by providing the missing files or configuration values.

If LEDBlinky is not displaying any errors, but it's still not working as expected, the first place to look for help is the [LEDBlinky Support](#) page.

Next, try using the LEDBlinky Troubleshooting application. The troubleshooting app (LEDBlinkyTroubleshooter.exe) will attempt to provide solutions for common issues such as the wrong buttons lighting up during game play. The app is self-explanatory and may provide a solution to your issue.

As another option, you can post questions to the Software group on the [ArcadeControls.com](#) forum. The current LEDBlinky support thread can be found [here](#). If you start a new thread, please include the word "LEDBlinky" in the subject.

To diagnose complex problems, LEDBlinky can generate a *Debug.log* file. Debug mode can be enabled from the LEDBlinky Configuration Application on the "Misc Options" tab. Checking the "Verbose" option will log even more information and generate additional *Debug_*.log* files – all of which will be zipped into the *Debug.zip* file. Using the Debug mode may degrade performance and should only be enabled to actively diagnose a problem.

If you wish to contact me directly, please post a personal message to "[arzoo](#)" on the ArcadeControls.com forum. You can also reach me via email at arzoo@LEDBlinky.net.

If you would like my help diagnosing a problem, please follow these steps:

- 1) From the LEDBlinky Configuration app, check the "Enable Debug Log" and "Verbose" options on the Misc Options tab.
- 2) Start your front-end and run through the process that's not working – like start a few games. Then quit your front-end.
- 3) Locate the *Debug.zip* file in the LEDBlinky folder and email it to me along with an explanation of what's not working.
- 4) Don't forget to turn off the "Enable Debug Log" option when you're done.

Running LEDBlinky in Stand-Alone (Command Line) Mode

Any front-end that supports the ability to launch an external application and pass parameters (e.g. Rom name) can use LEDBlinky. The LEDBlinky stand-alone program file is *LEDBlinky.exe*.

If your FE cannot launch an external application prior to starting a game, you can still use LEDBlinky with MAME. From the LEDBlinky configuration app, check the “Use MAME to Trigger the Game Start/Stop Events” option on the “MAME Config” tab. Then launch LEDBlinky with the FE Start parameter (see below) prior to starting your FE. You can even use this option without any FE - just using MAME or any variant that supports MAME output messages.

You should only enable the LEDBlinky features for events that your front-end supports. For example, if your front-end can launch *LEDBlinky.exe* when the front-end first starts, then you can use the “FE Startup Animation” and other Startup options. If your front-end can launch *LEDBlinky.exe* when it starts and stops a screensaver, then you can use the LEDBlinky screensaver options.

The LEDBlinky application can be launched multiple times (once for each event parameter). Don’t worry, only one instance will remain in system memory.

LEDBlinky command line syntax:

FE Start

```
LEDBlinky.exe 1
```

FE Quit

```
LEDBlinky.exe 2
```

Game Start

```
LEDBlinky.exe <rom>
```

```
LEDBlinky.exe <rom> <emulator>
```

Note: Use double quotes around parameters if the parameter values may include spaces.

Note: If you start a game without specifying the emulator, MAME is the default.

Game Stop

```
LEDBlinky.exe 4
```

Screensaver Start

```
LEDBlinky.exe 5
```

Screensaver Stop

```
LEDBlinky.exe 6
```

List Change

LEDBlinky.exe 8

Animation Start

LEDBlinky.exe <animation>.lwax

LEDBlinky.exe random.lwax

Add 'S' or 'SingleLoop' parameter to run through the animation sequence one time.

LEDBlinky.exe <animation>.lwax SingleLoop

LEDBlinky.exe random.lwax SingleLoop

For Audio Animations, the second parameter specifies the animation file used for the first frame.

LEDBlinky.exe audio.lwax <animation>.lwax

Note: Animation files must reside in the LEDBlinky\lwa folder.

Animation Stop

LEDBlinky.exe 11

Load MAME Controller File

LEDBlinky.exe 12 <controller file>

Note: If the filename is specified without the full path, the controller file must reside in the MAME cfg folder.

Note: LEDBlinky will only load values from the controller file that match the MAME defined XML schema (the same as any MAME cfg file).

Note: Input mappings defined in the default.cfg or <rom>.cfg files will take precedence over any mappings defined in the controller file.

Reset Ultrastik 360 (u360)

LEDBlinky.exe 13


LEDBlinky.exe 13 <map name>


Note: The Map Name refers to the .um files in the /jdr folder. Typical values: analog, joy2way, joy4way, joy8way, joydiag, vjoy2way. Default is joy8way.

Note: If no u360 joysticks are connected when LEDBlinky first starts, this command can be used to rescan/reinitialize the u360 joystick(s) – for example, when hot-swapping control panels.

LEDBlinky Core Application

The LEDBlinky core application runs in the Windows system tray. It accepts commands using command-line parameters (see [above](#)) or via a plug-in interface.

To display the LEDBlinky version and registration information, right-click on the tray icon  and select “About LEDBlinky”.

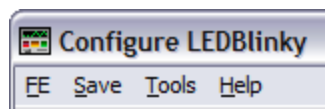
Should you wish to close the LEDBlinky application, right-click on the tray icon  and select “Exit LEDBlinky”.

The LEDBlinky application can be launched multiple times but only one instance will remain in system memory.

LEDBlinky Configuration Application

This application allows you to configure the various features of the LEDBlinky plug-in and provides quick access to the other LEDBlinky tools.

Menus



FE


Select your Front-End. Current options are “MaLa”, “AtomicFE”, “GameEx”, “HyperSpin”, “Maximus Arcade”, and “Other”. The MaLa tab, AtomicFE tab, GameEx tab, HyperSpin tab, and Maximus tab will be visible or hidden depending on the front-end selected.

Save


Save the current configuration. The LEDBlinky configuration file is *settings.ini* located in the LEDBlinky folder. The “Save” menu option is only enabled when one or more options have been modified. If you attempt to close the Configuration app prior to saving, you will be prompted to save the data.

Tools

Generate Input Map

This button runs the Generate LEDBlinky Input Map tool . You can also run this app directly - *GenLEDBlinkyInputMap.exe*. Before you can use the LEDBlinky plug-in, you must create an input map. A full description on how to use this application is provided in a later [section](#) of this document.


Controls Editor

This button runs the LEDBlinky Controls Editor . You can also run this app directly – *LEDBlinkyControlsEditor.exe*. A full description on how to use this application is provided in a later [section](#) of this document.

LED Animation Editor

This button runs the LEDBlinky Animation Editor . You can also run this app directly - *LEDBlinkyAnimationEditor.exe*. See the *AnimationEditor.pdf* file for detailed instructions on using the animation editor.

MAME Output Test

This button runs the MAME Output Test application . Use the MAME Output Test to view which outputs are generated by a MAME game and which controls (if any) are associated with the outputs.

Start the test app, then start MAME or MAME32, then start any game – the outputs and their associated controls will be listed whenever their state changes; On/Off.

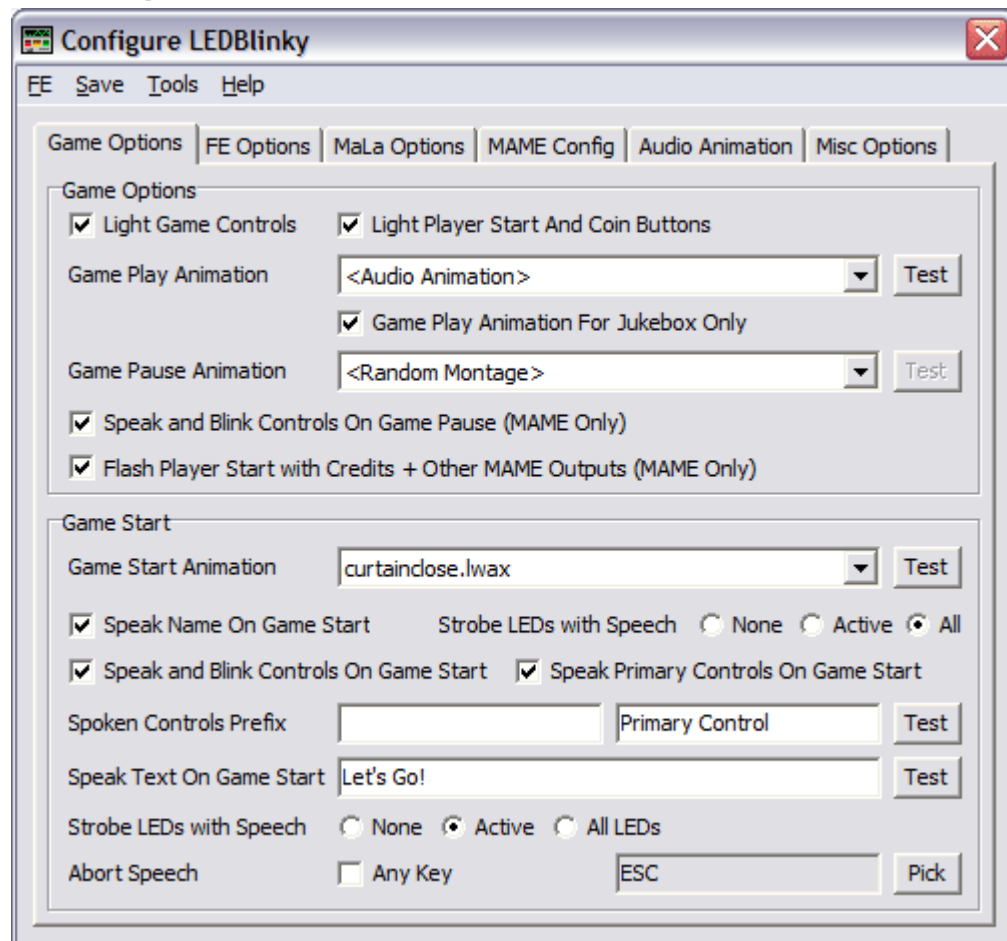
Any output can be added to the *MameOutputs.ini* file. To define outputs for a specific game, create an output section with the section key equal to the rom. For example, [digdug]. The [default] outputs will be used for any game which does not have its own output section. See the *MameOutputs.ini* file for a full description of the output key=value format.

Warning: You must start the test app prior to starting MAME or MAME32. If you start MAME or MAME32 first, it will most likely crash when the test app is started.

Help

Launches the LEDBlinky help documentation – this document!

Game Options



Light Game Controls

Check this box if you wish to light the controls used by MAME or any other emulator. You must also generate a Control-Color Profile for MAME and/or Other Emulators.

Light Player Start And Coin Buttons

Check this box if you wish to light the player Start and Coin buttons during game play and when using the MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade Demo Game Controls feature.

The *controls.ini* file does not provide the Start and Coin controls. With this option, Start and Coin controls will be included for each game. For example, a two player game will include Start1, Start2, Coin1, and Coin2.

Note: With this option enabled, any Start or Coin controls added from the LEDBlinky Controls Editor will be ignored.

Game Play Animation

Select the file name you wish to run when playing a game. The animation file list is populated with all `.lwa` files in the `/lwa` folder. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When `<Random>` is selected, a different file (from the list) will run each time a game is started. When `<Random Montage>` is selected, animations will be selected from the `/lwa` folder and play a single loop one after the other - in effect randomly stringing all the animations together. When `<Audio Animation>` is selected, the actual animation along with other audio options must be selected on the [Audio Animation](#) tab.

Note: If the “Light Game Controls” option is checked, the animation will only affect non-active controls (controls not used by the selected game).

LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. LED Animation files can also be generated using the LEDBLinky Animation Editor or manually.

Game Play Animation For Jukebox Only

Check this option if you only want the Game Play Animation to run for emulators or games designated as a jukebox application. Use the Controls Editor [Jukebox](#) option to designate which Emulator or ROM/Game is a jukebox application.

Note: Any ROM/Game control group (including an emulator default control group) can be designated as a jukebox application for the purpose of this option. For example, let's assume you are using an audio animation for Game Play and the Game Play Animation For Jukebox Only option is checked. Using the Controls Editor you have designated your jukebox application. If you also have another game (not a jukebox) and you want the audio animation to run for that game, then just designate it as a jukebox!

Game Pause Animation

Select the file name you wish to run when pausing (P) a MAME game. The animation file list is populated with all `.lwa` files in the `/lwa` folder. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When `<Random>` is selected, a different file (from the list) will run each time MAME is paused. When `<Random Montage>` is selected, animations will be selected from the `/lwa` folder and play a single loop one after the other - in effect randomly stringing all the animations together. When `<Audio Animation>` is selected, the actual animation along with other audio options must be selected on the [Audio Animation](#) tab.

Note: This feature only works for MAME version .118 or later.

If the “Speak and Blink Controls on Game Pause” option is checked, the game pause animation will start running after the control blinking is complete.

LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. LED Animation files can also be generated using the LEDBLinky Animation Editor or manually.

Speak and Blink Controls on Game Pause (MAME Only)

Check this box if you wish blink each button and have the Windows Text To Speech Synthesizer speak the button "action" when pausing (P) a MAME game. If you press pause a second time while the speech/blinking is still active, the feature will stop and game play will resume.

Note: This feature only works for MAME version .118 or later.

Flash Player Start with Credits + Other MAME Outputs (MAME Only)

Check this box if you wish to use MAME Outputs. MAME Outputs are game dependent – some games use them and some don't. The most common outputs are LED0 and LED1 which are most often used to flash the player start buttons when credits are available.

LEDBlinky uses the *MameOutputs.ini* file to map outputs to controls or device ports. By default, LED0 and LED1 are mapped to Start1 and Start2 – if your player start buttons have LEDs, they will flash when credits are available for supported games (for example; Asteroids). Any other outputs you wish to use must be manually added to the *MameOutputs.ini* file.

The following MAME outputs are most useful when used in conjunction with the LEDBlinky plugin: LED0, LED1, LAMP0, LAMP1, LAMP2, ... LAMPn. To determine which (if any) outputs are generated by a game, use the MAME Output Test application (*MameOutputTest.exe*). Start the test app from the Tools tab, then start MAME (or MAME32), then start any game – the outputs and associated controls will be listed whenever their state changes; On/Off.

Any output can be added to the *MameOutputs.ini* file. To define outputs for a specific game, create an output section with the section key equal to the rom. For example, [digdug]. The [default] outputs will be used for any game which does not have its own output section. See the *MameOutputs.ini* file for a full description of the output key=value format.

Note: This feature only works for MAME version .112 or later.

Game Start Animation

Select the file name you wish to run before the selected game starts. When <Random> is selected, a different file (from the list) will run each time a game starts. Only a single loop of the LED animation will be played (one pass through the file). The animation file list is populated with all .lwax files in the /lwa folder. LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. Click the "Test" button to run the selected animation, and click the button a second time to stop the animation.

Speak Name On Game Start

Check this box if wish to have the Windows Text To Speech Synthesizer speak the game name when starting a game.

Strobe LEDs with Speech

Select which LEDs (None, Active for selected game, or All) will blink as the Game Name is spoken. The LEDs will blink on an off based on the phonetics of the speech.

Speak and Blink Controls On Game Start

Check this box if you wish blink each button and have the Windows Text To Speech Synthesizer speak the button "action" when starting a game.

Speak Primary Controls On Game Start

Check this box if you wish to have the Windows Text To Speech Synthesizer speak the primary control(s) when starting a game. The primary controls for each MAME game are defined in the *controls.ini* file by the "P1Controls" tag.

P1Controls example:

```
P1Controls=8-way Triggerstick+joy8way+P1_BUTTON1|Spinner+dial
```

In this example (for Tron), the primary controls are an 8-way joystick and a spinner. LEDBlinky uses the values to the left of the plus (+) for the spoken text. For Tron the spoken text will be; "8-way Tiggerstick and Spinner".

Primary controls for other emulators are defined using the LEDBlinky Controls Editor. The Controls Editor can also be used to override the values in the *controls.ini* file (for MAME).

Spoken Controls Prefix

The Spoken Controls Prefix text will be spoken before the button "actions" are blinked and spoken. The Spoken Primary Controls Prefix text will be spoken before the primary controls are spoken (after the buttons). Click the "Test" button to hear how the text will sound using the current [Text To Speech](#) Options.

Speak Text On Game Start

Enter any text you wish to have spoken prior to starting a game. Click the "Test" button to hear how the text will sound using the current [Text To Speech](#) Options.

Strobe LEDs with Speech

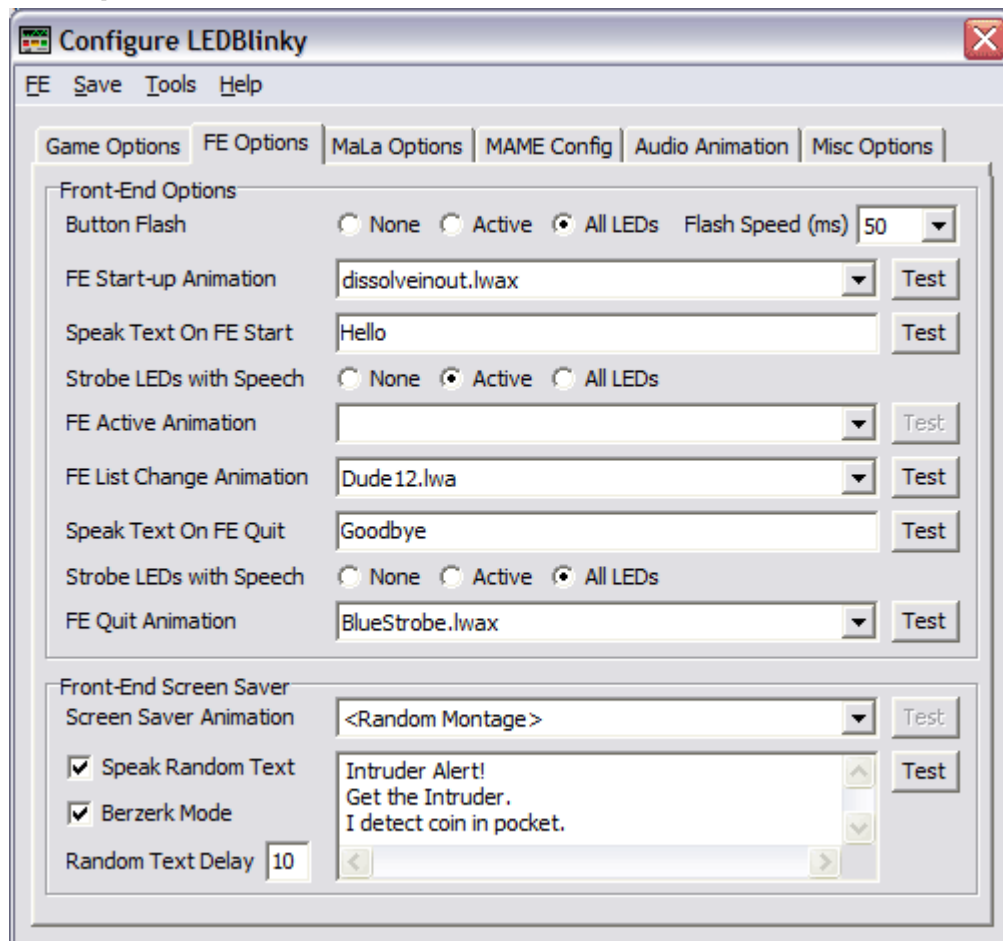
Select which LEDs (None, Active for selected game, or All) will blink as the Game Start message is spoken. The LEDs will blink on an off based on the phonetics of the speech.

Abort Speech

You may abort the game start speech (game name, button actions, primary controls, etc.) by pressing any button or a selected button. You may also abort the MAME pause speech (button actions).

To select a specific Abort Speech button, uncheck “Any Key” and click the “Pick” button, then press any key on your keyboard or control panel.

FE Options



Button Flash

Button Flash will do just that – flash your LEDs (one time) each time a button is pressed. Faster CPUs may require a longer Flash Speed – adjust to your preference.

FE Start-up Animation

Select the file name you wish to run when the front-end first starts. When <Random> is selected, a different file (from the list) will run each time the front-end is started. Only a single loop of the LED animation will be played (one pass through the file). The animation file list is populated with all *.lwax* files in the */wa* folder. LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Speak Text On FE Start

Enter any text you wish to have spoken when the front-end first starts. Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) Options.

Strobe LEDs with Speech

Select which LEDs (None, Active for selected game, or All) will blink as the front-end Start message is spoken. The LEDs will blink on an off based on the phonetics of the speech.

FE Active Animation

Select the file name you wish to run while the front-end is active (not during game play). When <Random> is selected, a different file (from the list) will run each time the front-end starts. The animation file list is populated with all `.lwax` files in the `/wa` folder. LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When <Audio Animation> is selected, the actual animation along with other audio options must be selected on the [Audio Animation](#) tab.

Note: This feature cannot be used in conjunction with the “Light MaLa Controls”, “Light AtomicFE Controls”, “Light GameEx Controls”, “Light HyperSpin Controls” or “Light Maximus Arcade Controls” option.

FE List Change Animation

Select the file name you wish to run each time the FE changes Emulators or Game lists. When <Random> is selected, a different file (from the list) will run each time the front-end is started. Only a single loop of the LED animation will be played (one pass through the file). The animation file list is populated with all `.lwax` files in the `/wa` folder. LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Note: This feature may not be fully supported by every front-end. For example, some front-ends may only play the list change animation when switching emulators, but not for game lists or category lists.

Speak Text On FE Quit

Enter any text you wish to have spoken when quitting the front-end. Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) Options.

Strobe LEDs with Speech

Select which LEDs (None, Active for selected game, or All) will blink as the front-end Quit message is spoken. The LEDs will blink on an off based on the phonetics of the speech.

FE Quit Animation

Select the file name you wish to run when quitting the front-end. When <Random> is selected, a different file (from the list) will run each time the front-end exits. Only a single loop of the LED animation will be played (one pass through the file). The

animation file list is populated with all `./wax` files in the `/wa` folder. LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Screen Saver Animation

Select the file name you wish to run when the front-end is in screensaver mode. The animation file list is populated with all `./wax` files in the `/wa` folder. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When `<Random>` is selected, a different file (from the list) will run each time the front-end enters screensaver mode. When `<Random Montage>` is selected, animations will be selected from the `/wa` folder and play a single loop one after the other - in effect randomly stringing all the animations together. When `<Audio Animation>` is selected, the actual animation along with other audio options must be selected on the [Audio Animation](#) tab.

LEDBlinky can run any LWAX or LWA file that conforms to the defined standards. LED Animation files can also be generated using the LEDBlinky Animation Editor or manually.

Speak Random Text

Enter any text you wish to have spoken when the front-end is in screensaver mode. One line will be spoken randomly at predefined intervals (Random Text Delay). Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) Options.

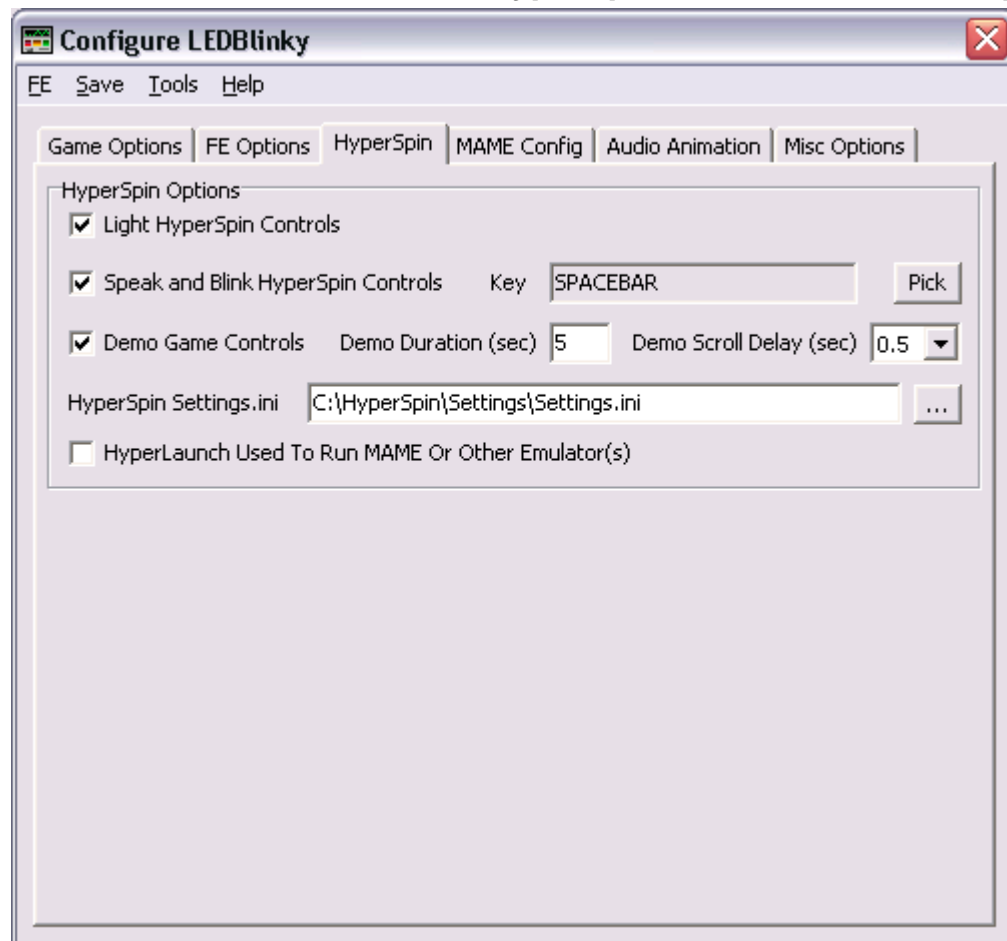
Berzerk Mode

With Berzerk Mode enabled, the speech rate will be randomly varied as each line of random text is spoken. For the best effect, use the Microsoft Sam voice which sounds similar to the original Berzerk voice!

Random Text Delay (seconds)

Enter the interval in seconds between random text messages. For example, a value of 60 will play a line of random text every minute while the screen saver is active.

MaLa / AtomicFE / GameEx / HyperSpin / Maximus Arcade Options



Light MaLa Controls

Light AtomicFE Controls

Light GameEx Controls

Light HyperSpin Controls

Light Maximus Arcade Controls

Check this box if you wish to light the controls used by the MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade user interface. You should also generate a Control-Color Profile for MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade.

Note: This feature is only useful if you have MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade controls mapped to buttons with LEDs.

Note: For GameEx, you must have “Enable Custom Keyboard Controls” = Yes in the GameEx Input settings.

Note: This feature cannot be used in conjunction with the “FE Active Animation” option.

Demo Game Controls

Check this box if you wish to briefly light the controls used by each game as you scroll through the MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade game lists.

Demo Duration (seconds)

The demo duration determines how many seconds a game's controls will be lit while scrolling through the game list. For example, if the demo duration is set to 5 and you stop scrolling on Asteroids, the Asteroids buttons will light up for 5 seconds, after which the buttons return to the FE controls or turn off (depending on the selected feature).

Demo Scroll Delay (seconds)

The demo scroll delay is the time between when you stop scrolling and when the game's controls light up. When set to zero, the game's controls will light immediately as you stop on each game, but this may affect the scroll performance on slower CPUs.

Speak And Blink MaLa Controls**Speak And Blink AtomicFE Controls****Speak And Blink GameEx Controls****Speak And Blink HyperSpin Controls****Speak And Blink Maximus Controls**

Check this box if you wish blink each button and have the Windows Text To Speech Synthesizer speak the button "action" when the FE is active.

To select the button (key) to active this feature, click the "Pick" button, then press any key on your keyboard or control panel. The same button (key) can also be used to abort the speech.

HyperSpin Settings.ini

For HyperSpin you must select the location of the *settings.ini* file (usually located in the HyperSpin *Settings* folder).

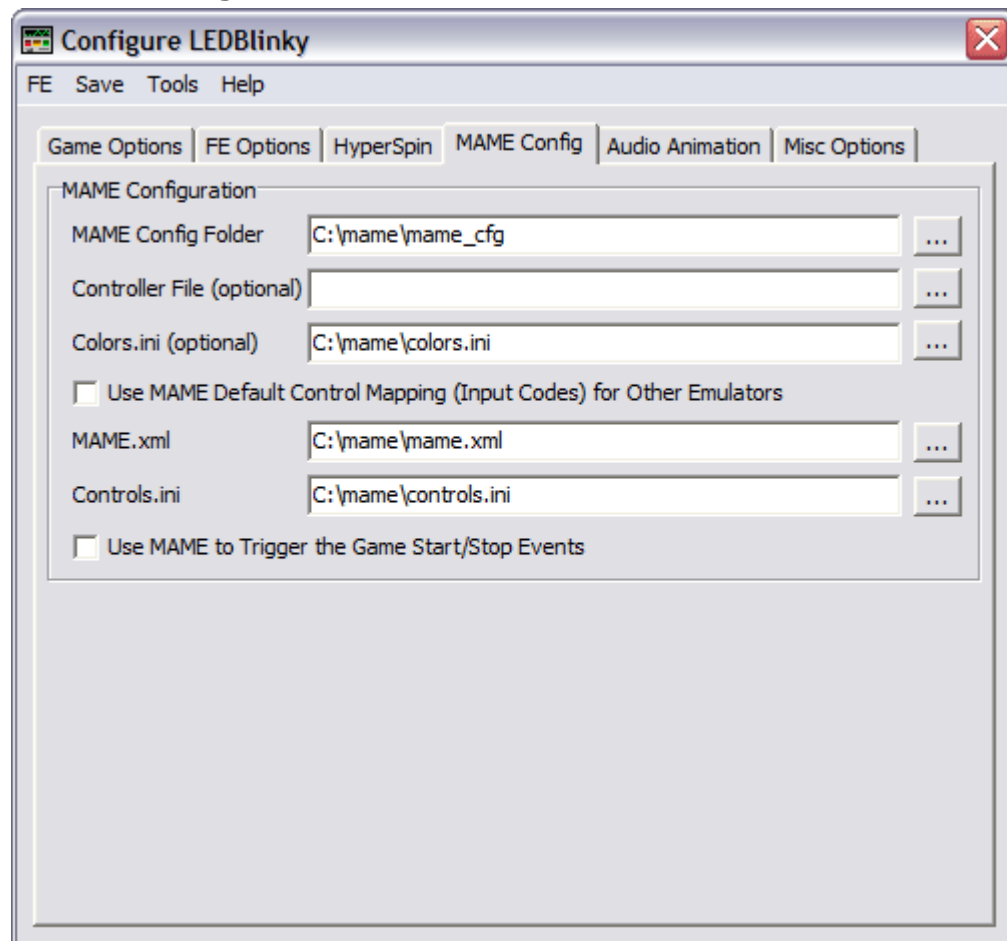
HyperLaunch Used To Run MAME Or Other Emulator(s)

When using HyperLaunch (version 2.0 or later) with HyperSpin to run one or more emulators (including MAME), this option should be checked. With this option enabled, LEDBlinky will ignore any invalid "Game Quit" commands sent from HyperSpin while HyperLaunch is active (in memory).


Maximus Default.ini

For Maximus Arcade you must select the location of the *default.ini* file (usually located in the Maximus Arcade *Preferences* folder).

MAME Config



MAME Config Folder

Click the folder browse button  to select the MAME config folder. This is usually the *cfg* folder. You must specify this folder to load MAME controls.

Controller File

Controller files are usually supplied by Control Panel vendors (xArcade, SlikStik, etc.). They provide the default input controls for their layout. If you use a controller file when running MAME, set the path to that file.

Colors.ini

Optionally, you can set the path to the *colors.ini* file. The *colors.ini* file is used to set game specific control colors (or intensities). A version of this file with authentic control panel colors supporting over 1200 ROMs (at last count) has been included with LEDBlinky. An updated version may be available for download from the ArcadeControls forum - Thanks headkaze!

Note: Colors defined for an individual ROM/Game using the LEDBlinky Controls Editor will override the values in the *Colors.ini* file for that ROM/Game.

If you wish to add additional entries to the *Colors.ini* file, valid colors are defined in the *Color-RGB.ini* file. Valid intensities are 0 – 48, or the following built-in blinking effects;

129 = Ramp Up / Ramp Down

130 = On / Off

131 = On / Ramp Down

132 = Ramp Up / On


Use MAME Default Control Mapping for Other Emulators

Unfortunately, there is no way for LEDBlinky to know the control input codes (keycodes) for each non-MAME emulator or individual game. In this regard, you have two choices – define the input codes manually using the [LEDBlinky Controls Editor](#), or use MAME's control-input map for all other emulators (this option).


If your control panel always uses the same buttons mapped to the same input codes (keycodes) for all emulators including MAME, then you may wish to check this option. Remapping any buttons in MAME for 'All Games' will also be reflected for other emulators. If any of your emulators use control mapping that differs from MAME, then do not use this option.

Note: This only affects the buttons that LEDBlinky lights up. The actual keycode assignments must be changed for each individual emulator via its configuration features.

MAME.xml

Click the file browse button  to select the *MAME.xml*. This file is generated by MAME using the `-listxml` command. You must specify this file to light MAME controls. This field is not visible (and not required) when using MaLa.

Controls.ini

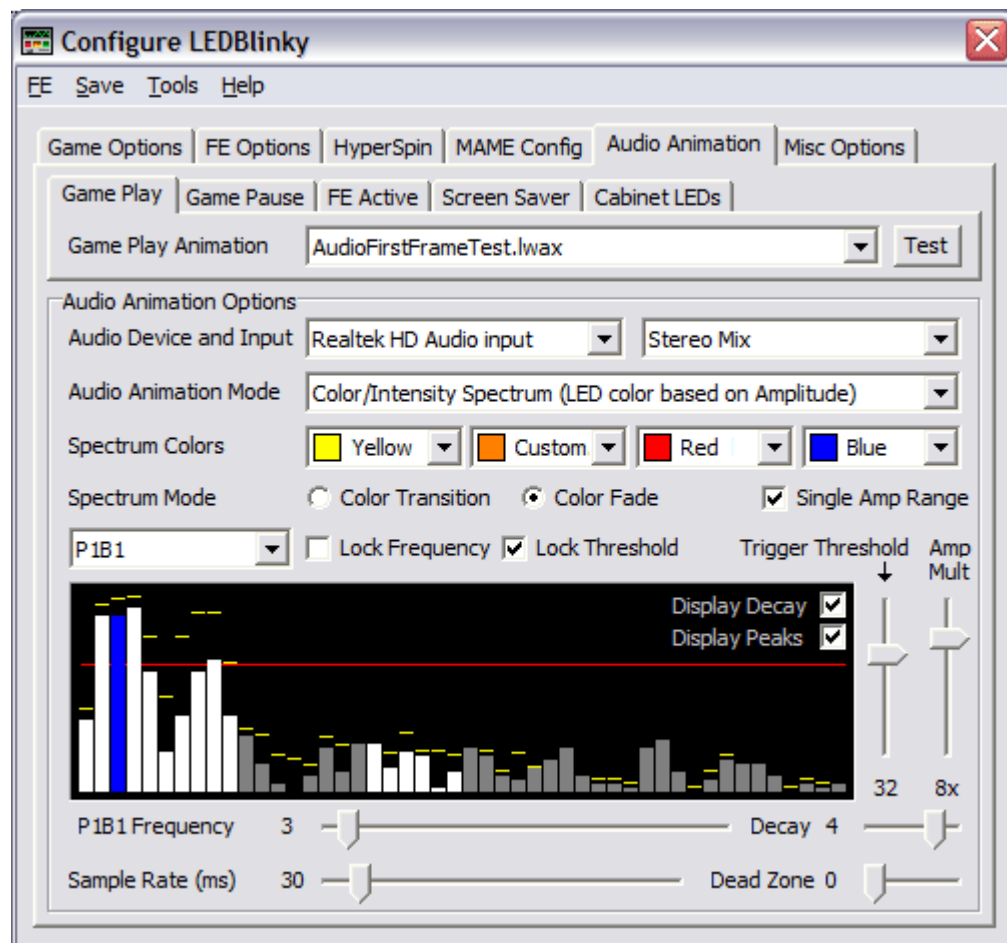
Click the file browse button  to select the *Controls.ini* file. You must specify this file to light MAME controls. If you don't have a controls.ini file, it can be downloaded from controls.arcadecontrols.com/controls. This field is not visible (and not required) when using MaLa.

Use MAME to Trigger the Game Start/Stop Events

With this option checked, MAME output messages will be used to detect when a game (MAME only) is started and stopped. You should only use this option if your front-end cannot launch LEDBlinky prior to launching a game.

You can also use this option to launch LEDBlinky without any front-end, just using MAME or any MAME variant that supports MAME output messages.

Audio Animation



Audio Animations can be active when a game is played and/or paused, and/or when the FE is active, and/or when the FE screen saver is active, and/or for Cabinet LEDs.

Audio Animations are enabled by selecting <Audio Animation> from the following animation file drop-down lists:

- “Game Play Animation” on the “Game Options” tab.
- “Game Pause Animation” on the “Game Options” tab.
- “FE Active Animation” on the “FE Options” tab.
- “Screen Saver Animation” on the “FE Options” tab.
- Cabinet “Animation File” on the “Misc Options” tab in the Cabinet LEDs section.

Note: If you are using **Vista** as your operating system, you may need to enable the “Stereo Mix” or “What You Hear” recording device to use the LEDBlinky audio animation features. Here’s how:

1. Right click the volume icon in the system tray and select “Recording Device”. Or you can open the “Sounds” dialog from the control panel.

2. On the Sound “Recording” tab, right click on any blank area and select “Show Disabled Devices”. This should then add a device that may be called “Stereo Mix” or “What You Hear”.
3. Right click the new device and select “Enable”.

Game Play Animation**Game Pause Animation****FE Active Animation****Screen Saver Animation****Cabinet LEDs Animation**

Select the animation file you wish to use for the Audio Animation.

Note: The “Blink Controls” animation mode only uses the first frame of the animation.

Note: No animation file is required for the “Color/Intensity Spectrum” animation mode.

Note: All Cabinet LED animations share the same audio animation.

Test

Click the “Test” button to start the audio animation testing with the selected options. Most options can be changed while actively testing. Click the button a second time to stop the test.

Note: You must have active audio input for the test to work – play some music!

Audio Device

Select the sound card used for the audio input.

Audio Input

Select the audio input source. Options here are dependent on the selected audio device (sound card).

Audio Animation Mode

Select the primary audio animation mode you wish to use. Other audio options may be enabled or disabled depending on which animation mode is selected. The following animation modes are available:

Pulse Animation

Using a selected frequency, the specified animation file will be advanced through the frames based on a trigger threshold or increasing amplitude. Basically, the animation can advance to the beat of the music.

Color/Intensity Spectrum

In this mode the color of each LED fades or transitions based on the frequency amplitude. Each control (LED) can be assigned to a different frequency. For RGB LEDs,

a common color range can be defined. Basically, the color or brightness changes as the music gets louder or softer.

Blink Controls

Controls (LEDs) are toggled on or off based on a trigger threshold. Each control (LED) can be assigned to a different frequency and trigger threshold. For RGB LEDs, the color is set by the first frame of the selected animation - this allows you to define custom color layouts. Basically, the controls blink on or off based on a volume threshold.

Amplitude Animation

This mode sequentially assigns each frame of the animation to an amplitude. For example, amplitude level 1 uses frame 1, amplitude level 2 uses frame 2, amplitude level 3 uses frame 3, etc. By assigning a different frequency to each button, you can effectively design an animation that sets the color for every button at every amplitude value. This mode works best with animations that have approximately 50 frames.

Spectrum Colors

The Spectrum Colors lists allow you to define one, two, three, or four colors for use with the Color/Intensity Spectrum animation mode. For all other modes, these fields are disabled. To specify a color not in the list, select "Custom". Colors change left to right from low amplitude to high amplitude. See Spectrum Mode below for more details.

Note: For non RGB LEDs the intensity (brightness) will increase regardless of which colors are selected.

Note: The Spectrum Colors options are only enabled for the Color/Intensity Spectrum animation mode.

Spectrum Mode – Color Transition

When using the Color Spectrum animation mode, the Color Transition option uses the amplitude to transition between each adjacent color. Colors transition left to right from low amplitude to high amplitude. For example, if you select "Blue", "Yellow", "Red", each control (LED) will transition from blue to yellow and then yellow to red as the amplitude increases.

Spectrum Mode – Color Fade

When using the Color Spectrum animation mode, the Color Fade option uses the amplitude to fade up from one color to the next. Colors fade left to right from low amplitude to high amplitude. For example, if you select "Blue", "Magenta", "Red", "Red"; each control (LED) will fade up blue, then magenta, then red as the amplitude increases. Blue for the first 25%, magenta for the next 25%, and red for the last 50% (since Red was selected twice).

Note: The Spectrum Mode options are only enabled for the Color/Intensity Spectrum animation mode.

Single Amp Range

With this option enabled, the intensity (brightness) of each LED will increase from 0 to 100% across the color breakpoints. For example, if you have two colors selected, “Yellow” and “Red” – Yellow will light from 0 to 50% intensity, and then Red will light from 50% to 100% intensity. With this option disabled, Yellow will light proportionally from 0 to 100%, and then Red will light proportionally from 0 to 100%.

Note: The Single Amplitude Range option is only enabled for the Color/Intensity Spectrum animation mode with the Spectrum Mode set to “Fade”.

Trigger Mode – Trigger Threshold

The Trigger Threshold option will pulse an animation or toggle (On/Off) the control based on a specific amplitude. Each control (LED) can be assigned to a different frequency and trigger threshold.

Use the Trigger Threshold slider on the right side of the Spectrum display to set the threshold. When actively testing, the selected threshold is displayed as a red horizontal line above each frequency. You can use the Dead Zone slider to reduce the sensitivity.

Trigger Mode – Increase Amplitude

The Increase Amplitude option will pulse an animation or toggle (On/Off) the control each time the amplitude changes from decreasing to increasing. This usually causes rapid animation pulses or control blinking. You can use the Dead Zone slider to reduce the sensitivity.

Note: The Trigger Mode options are not enabled for the Color/Intensity Spectrum animation mode.

Control Drop-down List (left side, above Spectrum display)

This drop-down list includes all Controls defined for your control panel. Selecting a control does not set any specific option; rather it allows you to assign other options (Frequency or Trigger Threshold) to that control.

For example, if you wish to set player one button one to the third frequency band, select “P1B1” from the drop-down list, then set the Frequency slider to value 3. The frequency and threshold value for each control will be retained regardless of which control is displayed.

Note: The Control list is not enabled when both Lock Frequency and Lock Threshold are checked.

Lock Frequency

Check Lock Frequency to use a single frequency across all controls (LEDs).

Note: Lock Frequency is not enabled for the Pulse Animation mode.

Lock Threshold

Check Lock Threshold to use a single trigger threshold across all controls (LEDs).

Note: Lock Threshold is only enabled for the Blink Controls animation mode.

Trigger Threshold

The Trigger Threshold determines when the animation will be pulsed (advanced) or when the control's LED will toggle on or off. Use the Trigger Threshold slider to adjust the trigger amplitude for the current control or all controls (see Lock Threshold).

When actively testing, the selected threshold is displayed as a **red** horizontal line above each frequency.

When Lock Threshold is unchecked, The Threshold slider label will display the name of the current control, otherwise the label will display "Trigger Threshold".

Note: The Trigger Threshold slider is not enabled for the Color/Intensity Spectrum animation mode or when the Increase Amplitude option is enabled.

Amp Mult

Use the Amplitude Multiplier slider to adjust the overall input amplitude. This adjustment depends on your PC and audio player's volume setting. Increasing the Amplitude Multiplier may cause low frequency amplitude values to occasionally clip (max out at the highest color/intensity).

Slide the Amplitude Multiplier to its lowest position to set "auto" mode. In auto mode the software will increase or decrease the multiplier dynamically to maintain an optimal value. The *settings.ini* file contains three values which effect how the auto Mult Amp mode works. The default values for these settings should be adequate – but they can be modified if you wish.

- AutoAmpMultUpdateRate – The rate at which the multiplier value is adjusted in milliseconds. The default value is 2000 (2 seconds).
- AutoAmpMultTargetAvgLow and AutoAmpMultTargetAvgHigh – The target amplitude range for which the multiplier remains stable. These are average amplitude values between 1 and 50. The defaults are 8 and 14. The average amplitude is calculated using the first 10 frequencies only.

Note: You should leave your PC's volume at a fixed level and use your speaker's volume control to adjust the music output level.

Display Decay

With the Display Decay option unchecked, the spectrum analyzer will display the amplitude drop-off (decay) in real-time. With the option checked, the amplitude drop-off will use a linear delay based on the value set by the Decay slider. This has the effect of smoothing the spectrum analyzer display.

Note: This option only affects the spectrum analyzer display – it has no effect on the LEDs.

Display Peaks

With this option checked, additional horizontal segments will display above each frequency (during active testing) to help visualize the amplitude peaks (momentary high values).

Note: This option only affects the spectrum analyzer display – it has no effect on the LEDs.

Trigger Frequency

The Trigger Frequency determines which frequency band is used to pulse (advance) the animation or blink the control's LED. Use the Trigger Frequency slider to select the frequency band for the current control or all controls (see Lock Frequency).

When actively testing, the selected frequency band is displayed as a **blue** or **red** bar. All other frequency bands assigned to controls are displayed as white bars. Any unassigned frequency bands are displayed as **gray** bars.

When Lock Frequency is unchecked, The Frequency slider label will display the name of the current control, otherwise the label will display "Trigger Frequency".

Decay

The Decay slider introduces a linear delay as the LED intensity drops off. This has the affect of smoothing the color/intensity changes. Higher values have less affect (less smoothing). Setting the value to zero (0) disables the decay and causes the color/intensity changes to occur in real-time (fast).

Note: The Decay slider is not enabled for the Pulse Animation mode.

Sample Rate (ms)

Use the Sample Rate slider to adjust how often (in milliseconds) the audio input is sampled. Higher values have the affect of slowing down the LED activity. The lowest value (10ms) will result in the LED activity most accurately responding to the audio input (music). PC's with less processing power may need to use a slightly higher value (30ms).

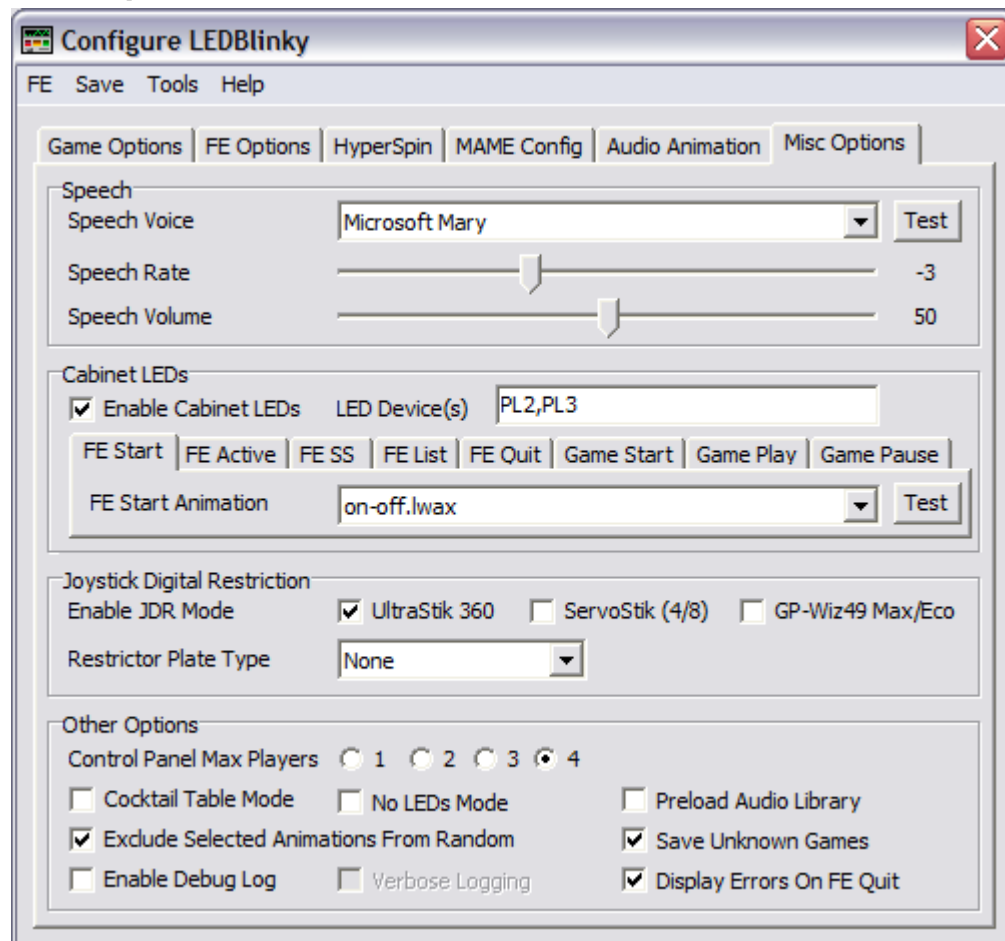
Dead Zone

The Dead Zone slider acts as a sensitivity adjustment. It is most effective when the "Increase Amplitude" option is enabled.

When actively testing, the selected threshold will be displayed as two **red** horizontal lines above each frequency – representing the top and bottom of the dead zone.

Note: The Dead Zone slider is not enabled for the Color/Intensity Spectrum animation mode.

Misc Options



Speech Voice

By default, the Windows Text To Speech Synthesizer comes with the Microsoft Sam voice. Microsoft also provides two other freely downloadable voices – Mary and Mike. Other third-party voices can be purchased and downloaded.

Speech Rate

Move the slider to set the voice speech rate from -10 to +10. The default is 0.

Speech Volume

Move the slider to set the voice speech volume from 0 to 100. The default is 100 (full volume).

Cabinet LEDs

LEDBlinky can run a secondary (independent) animation for cabinet or attract mode lighting. To use this feature, you must have additional LED devices (one or more) wired to the cabinet LEDs.

Enable Cabinet LEDs

Check this box to enable cabinet LED animations.

Note: If you are only lighting your control panel LEDs, do not enable the Cabinet LEDs option.

LED Device(s)

To use the secondary animation features, you must have additional LED output control devices (one or more) wired to the cabinet LEDs. Each LED device (LED-Wiz, Pac-LED64, Pac-Drive, or U-HID) is assigned a unique ID. Specify the device IDs used for cabinet lighting as a comma separated list. LED-Wiz IDs must be prefixed with “LW”. Pac-LED64 IDs must be prefixed with “PL”. Pac-Drive or U-HID IDs must be prefixed with “PD”.

For example, let's say you have a LED-Wiz (ID2) and two Pac-LED64s (IDs 1 and 2) wired to your cabinet LEDs. You would specify LW2,PL1,PL2.

Cabinet LEDs Event Tabs

Cabinet LED animations can run for various events. To enable an animation for each event, click the associated tab and then select an animation file.

Note: If the same animation is selected for consecutive events, LEDBlinky will continue the animation as the event(s) change without interruption. For example, if you want to run an uninterrupted animation while the front-end or emulator is active, select the same animation for the FE Active, FE Screen Saver, Game Play, and Game Pause events.

Cabinet LEDs FE Start and Quit Animation

Select the file name you wish to run on the cabinet LEDs when the front-end first starts and/or quits. When <Random> is selected, a different file (from the list) will run each time the front-end is started. Only a single loop of the LED animation will be played (one pass through the file). Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Cabinet LEDs FE Active Animation

Select the file name you wish to run on the cabinet LEDs while the front-end is active (not during game play). When <Random> is selected, a different file (from the list) will run each time the front-end starts. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When <Audio Animation> is selected, the Cabinet LEDs Animation along with other audio options must be selected on the [Audio Animation](#) tab.

Cabinet LEDs FE SS (Screen Saver) Animation

Select the file name you wish to run on the cabinet LEDs while the front-end screen saver is active. When <Random> is selected, a different file (from the list) will run each

time the screen saver starts. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When <Audio Animation> is selected, the Cabinet LEDs Animation along with other audio options must be selected on the [Audio Animation](#) tab.

Cabinet LEDs FE List (Change) Animation

Select the file name you wish to run on the cabinet LEDs each time the front-end changes Emulators or Game lists. When <Random> is selected, a different file (from the list) will run each time the list is changed. Only a single loop of the LED animation will be played (one pass through the file). Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Cabinet LEDs Game Start and Quit Animation

Select the file name you wish to run on the cabinet LEDs each time a game starts. When <Random> is selected, a different file (from the list) will run each time a game is started. Only a single loop of the LED animation will be played (one pass through the file). Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

Cabinet LEDs Game Play Animation

Select the file name you wish to run on the cabinet LEDs while playing a game (emulator is active). When <Random> is selected, a different file (from the list) will run each time a game is played. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When <Audio Animation> is selected, the Cabinet LEDs Animation along with other audio options must be selected on the [Audio Animation](#) tab.

Cabinet LEDs Game Pause Animation

Select the file name you wish to run on the cabinet LEDs while a MAME game is paused. When <Random> is selected, a different file (from the list) will run each time a game is paused. Click the “Test” button to run the selected animation, and click the button a second time to stop the animation.

When <Audio Animation> is selected, the Cabinet LEDs Animation along with other audio options must be selected on the [Audio Animation](#) tab.

Note: This feature only works for MAME version .118 or later.

Enable JDR Mode

Joystick Digital Restriction (JDR) allows digital joysticks such as an Ultimarc Ultrastik 360 or a 49-Way controlled via the Groovy Game Gear GP-Wiz49 Max/Eco to be configured as is appropriate for the active game (2-way, 4-way, 8-way, etc.). An Ultimarc ServoStik’s restrictor plate can also be switched between 4-way and 8-way.

When a game is started, the JDR mode is set based on the game's [primary control](#). When the front-end is active, the JDR mode is set to 4-way by default.

UltraStik 360

Check this box to enable JDR mode for any/all UltraStik 360 joysticks. Do not enable this feature if your control panel does not use any UltraStiks.

Each primary control type translates to an UltraStik 360 map file:

JOY2WAY = Joy2Way.um
VJOY2WAY = vJoy2Way.um
JOY4WAY = Joy4Way.um
JOY8WAY = Joy8Way.um
JOYDIAG = JoyDiag.um
STICK = Analog.um
DOUBLEJOY2WAY = Joy2Way.um
VDOUBLEJOY2WAY = vJoy2Way.um
DOUBLEJOY4WAY = Joy4Way.um
DOUBLEJOY8WAY = Joy8Way.um
TRACKBALL = Anoaalog.um
DIAL = Analog.um
PADDLE = Analog.um
LIGHTGUN = Analog.um
PEDAL = Analog.um
PEDAL2 = Analog.um

See [here](#) for a description and how to set a custom UltraStik map (.um file) for a specific ROM/Game.

Restrictor Plate Type

Ultimarc offers physical restrictor plates which can be installed into your UltraStik 360 joysticks. If you have installed a restrictor plate, select the type from the list. Each value will compress the outer range of the joystick. Custom-1 has the most compression and Custom-15 the least compression. Custom-9 corresponds to the physical 4-Way/8-Way/Circular restrictor plate.

ServoStik (4/8)

Check this box to enable automatic 4-way/8-way switching for any/all ServoStik joysticks. Do not enable this feature if your control panel does not use any ServoStiks.

The ServoStik restrictor plate will be rotated into 4-way mode for any game which uses a 4-way or 2-way joystick as its primary control. For all other games (or the front-end), the restrictor plate will be rotated into 8-way mode.

Control Panel Max Players

Set this value to the maximum number of players on your control panel for which you wish to light LEDs. This feature may prevent unexpected buttons from lighting up when your buttons are mapped to input codes that are normally used as defaults for other player buttons not available on your control panel.

For example, MAME uses (by default) keycodes I, J, K, and L for the player 3 joystick. If your control panel only supports two players and you use keycodes I, J, K, or L for your buttons, then these buttons may light incorrectly when you play a 4-player game. By setting the Control Panel Max Players to 2 for a two player control panel, you will avoid this problem.

Cocktail Table Mode

Enable Cocktail Mode if your cabinet is a cocktail table. Normally a multi-player *alternating* player game will only light the controls for player 1. With cocktail mode enabled, all player controls will light. For example, Galaxian will light the controls for Players 1 and 2.

Preload Audio Library

Enable Preload Audio Library when <Audio Animation> has not been selected for any animation feature and you are using an external command to start the audio animation. For example, your FE is using a pre-launch command or script to load the audio animation before launching a jukebox application.

No LEDs Mode

Enable No LEDs Mode ONLY when you are using LEDBlinky without any LED control devices. This is useful if you wish to use the LEDBlinky speech or Joystick Digital Restriction features but do not have any LEDs on your control panel.

Exclude Selected Animations From Random

If you have configured LEDBlinky to run a <Random> or <Random Montage> animation and also configured one or more other animation options to use a specific selected animation, then check this box to exclude the selected animation(s) from the random. For example, if the FE Start animation is "CurtainClose" and the Game Start animation is <Random>, then this option will prevent the CurtainClose animation from ever being selected randomly.

Enable Debug Log

With the Debug Log enabled, LEDBlinky will add debug messages to *Debug.log* file located in the *LEDBlinky* folder. This should only be used when diagnosing a problem.

With the Debug Log disabled, LEDBlinky will log all error messages to the *LEDBlinky.log* file.

Verbose Logging

LEDBlinky will log lots-o-messages to the debug file and additional *Debug_*.log* files. All debug files will be zipped into a *Debug.zip* file. This may slow LEDBlinky performance and should only be used when diagnosing a problem.

Note: Please include the *Debug.zip* file with any correspondence to diagnose problems.

Display Errors On FE Quit

LEDBlinky logs all errors to the *LEDBlinky.log* file in the *LEDBlinky* folder. If one or more errors occur during execution, LEDBlinky will display a message after you exit the front-end and provide you with the option to display the log file.

Game Specific Animations

LEDBlinky can run unique animations on a game-by-game basis. To use this feature, the animation files (.lwax only) MUST have the same name as the game or rom, and you must have a default animation selected for the related animation option in the Configuration app. Place the game specific animations in the desired "event" folder under the /LEDBlinky/GameSpecific/ folder.

The following events folders can trigger a game specific animation:

- Game Start
- Game Play
- Game Play using an audio animation
- Game Pause
- Game Pause using an audio animation
- Cabinet LEDs Game Start
- Cabinet LEDs Game Play
- Cabinet LEDs Game Play using an audio animation
- Cabinet LEDs Game Pause
- Cabinet LEDs Game Pause using an audio animation

Note: [Cabinet LEDs](#) must be enabled to use any of the cabinet LED events.

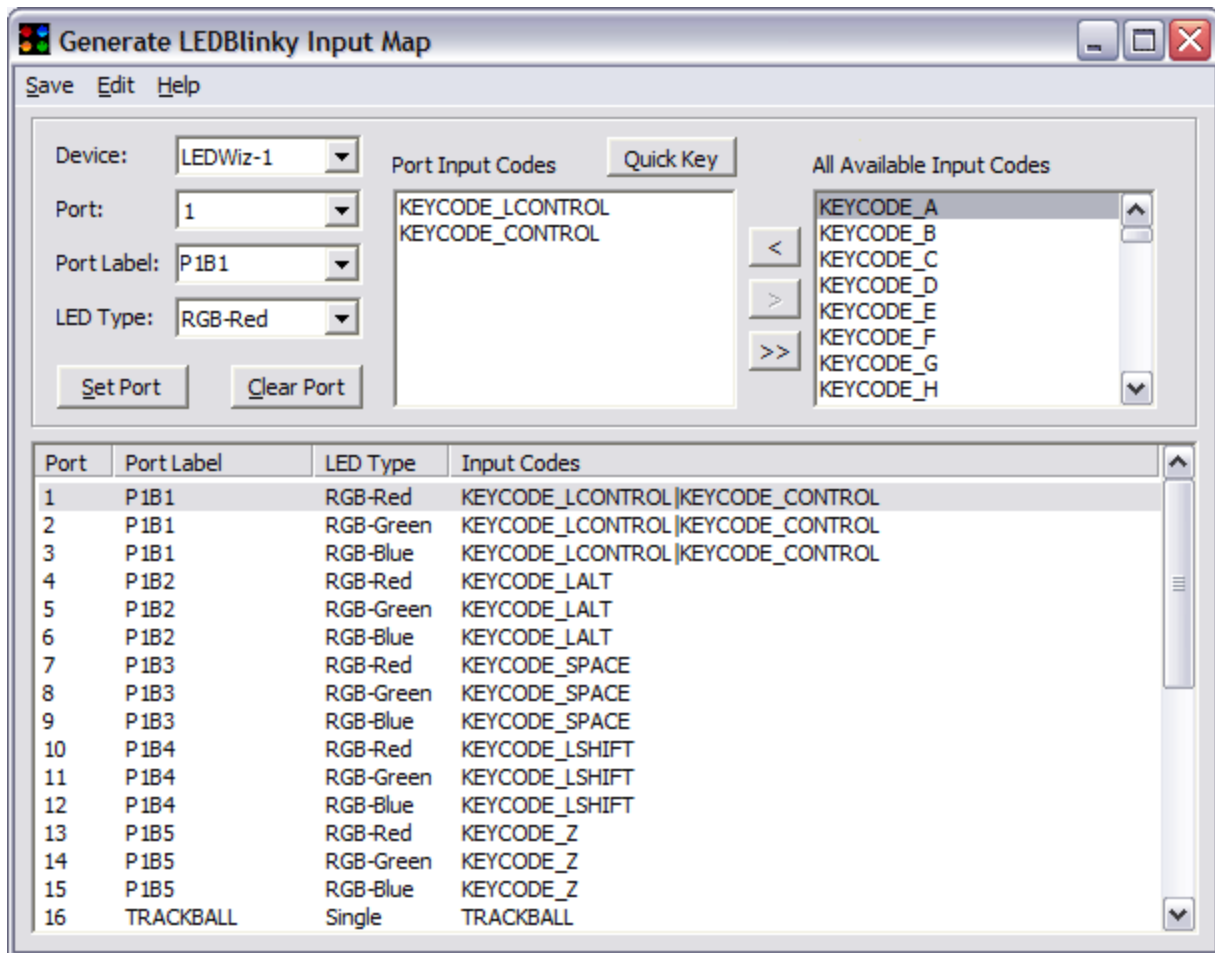
For example, let's say you wish to run a specific animation when playing Joust; first you must have an animation selected for the [Game Play Animation](#) option - this is the default animation. Next, create (or copy) your Joust animation into the \LEDBlinky\lwa\GameSpecific\GamePlay folder. The file MUST have the exact same name as the game/rom the emulator is running. So for Joust, you would use "joust.lwax". That's it.

Generate LEDBlinky Input Map Application

The input map defines the relationship between each wired port on your LED-Wiz and/or Pac-LED64 and/or Pac-Drive and/or U-HID device(s), and the keyboard input code for a button or input codes for other controls (Joysticks, Trackball, etc). This is a fixed relationship and should never change unless you rewire your LED device, or rewire your keyboard encoder, or reassign your keyboard encoder values.

Think about it this way – Each LED button on your control panel is physically wired to a port on the LED-Wiz, Pac-LED64, Pac-Drive, or U-HID. And each LED button is also physically wired to a port on your keyboard encoder and assigned a keyboard value. The input map file ties the LED device port to the assigned keyboard input code. LEDs for other controls are also wired to the LED device and these also have specific input codes.

The input map also defines a Port Label and LED Type for each wired port. The Port Labels serve two purposes, they tie together three ports for RGB LEDs, and they provide an easy reference to the ports from within the other LEDBlinky tools. The LED Type defines LEDs as Single or Red, Green, Blue for RGB.



Here's the general process for creating your input map. Each application feature is described in greater detail below.

1. Select a LED Device and Port. Optionally, you can click the port row on the lower list. The selected LED will light up on your control panel.
2. Select a Port Label from the list or type your own.
3. Select the LED Type (Single, Red, Green, or Blue).
4. Select one or more Input Codes from the list of all available Input Codes on the right. Optionally, you can click the "Quick Key" button and then press any key on the keyboard to assign the Input Code.
5. Click the "Set Port" button.
6. Repeat for next Port.
7. Click the "Save" button.

Device

The GrovyGameGear LED-Wiz and Ultimarc Pac-Drive devices have a hard-coded device id assigned at the time of purchase. The Ultimarc Pac-LED64 and U-HID devices has a firmware device id that can be assigned using Ultimarc software. You will need to set the input mapping for every wired port on all LED devices in your control panel.

Port

Pac-LED64 devices have 64 ports each, LED-Wiz devices have 32 ports each, and Pac-Drive / U-HID devices have 16 ports each. Your control panel may not use all ports on each LED device. A port can be selected from the drop-down list or by clicking on any port row in the lower pane.

Port Label

The port label serves two purposes – it ties together three ports for RGB LEDs, and it provides an easy reference to the ports from within the other LEDBlinky tools. You can select a port label from the drop-down list or enter your own label.

When specifying the port label for an RGB LED, you must use the same label for the red, green, and blue ports. Port labels for single LEDs must be unique.

LED Type

The LED type specifies that a port is wired to a single color LED, or the red, green, or blue lead for an RGB LED.

Port Input Codes

One or more input code can be assigned to each port. For LEDs that are illuminating a control that is wired to the keyboard encoder (usually buttons), you should only specify a single input code. For other controls not wired to the keyboard encoder (trackball, analog joysticks, spinners, etc.) you should specify all the input codes that could apply.

For example, in the sample screen shot above, five buttons have RGB LEDs, and the trackball has a single LED. Each of the buttons is assigned the same port label for all three colors (red, green, and blue), and the same input code. The trackball has its own unique port label and has been assigned the “TRACKBALL” input code. Analog controls such as a trackball can also be assigned analog input codes such as “MOUSECODE_1_ANALOG_X” and/or “MOUSECODE_1_ANALOG_Y”, etc.

Note: Assigning an input code to each port is not required if you are only creating LED animations for use with applications other than LEDBlinky.

Quick Key

Click this button to quickly select and assign a keyboard input code. When the button is depressed, it will flash ‘Press Key’; then press any key on the keyboard. The input code will be selected and added to the Port Input Codes list. Additionally, if the Port Label and LED Type have been specified and validated, the values will be assigned to the Port (the same as clicking the “Set Port” button). To abort the Quick Key mode, click the button a second time.

Note: The Quick Key feature cannot be used to assign joystick buttons – these must be selected manually from the list.

All Available Input Codes

This is the list of all possible input codes used by MAME. LEDBlinky also maps these input codes for use with MaLa, AtomicFE, GameEx, HyperSpin, or Maximus Arcade controls and other emulators.

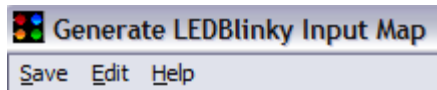
Set Port

Click this button to assign the Port Label, LED Type, and Input Code(s) to the selected Port. Validation will confirm that the port label is unique for a single type LED or that the port label and color is unique for an RGB LED.

Clear Port

Clears the selected port.

Menus



Save

Saves the Input Map!

Edit

The Edit menu provides options to copy and paste port values for RGB LEDs. This allows you to quickly add the three ports required for each RGB LED.

1. Select a blank port wired to a RGB and set the port information (Port Label, LED Type, and Input Codes), and then click "Set Port".
2. From the Edit menu, click "Copy Port". You can also right-click the port and select "Copy Port" from the pop-up menu.
3. Select the second blank port wired to the RGB. From the Edit menu, click "Paste Port" for the correct color. You can also right-click the port and select "Paste Port".
4. Repeat for the final blank port wired to the RGB.

Note: The "Copy Port" menu option is only enabled when an existing RGB port is selected. The "Paste Port" menu options are only enabled when a blank port is selected.

LEDBlinky Controls Editor

Use the LEDBlinky Controls Editor to configure the controls for any Emulator, Game, or Front-end. For each control you can specify the color or intensity, the voice 'action', and if necessary the input codes associated with the control. Controls can be defined as the default for an emulator, or individually for a specific game.

The primary need for the Controls Editor is to configure any non-MAME emulator or game. Since there is no way for LEDBlinky to know the control-input mapping (button assignments), or specific controls, or button colors for each non-MAME emulator, you must provide this information manually. Additionally, you can use the Controls Editor for MAME games to override values in the *Controls.ini* and *Colors.ini* files.

The Controls Editor provides a number of features to ease the configuration process. Each time you play an unknown game (one for which no unique controls are defined), the emulator and ROM/game name is stored. From the Controls Editor import menu, you can display the list of unknown games and select which you wish to import. You can then define the controls. Controls for player 1 can be copied to players 2, 3, and 4. If the controls for one game are similar to another, you can copy the entire ROM/Game.

The trick to getting your buttons to light for each emulator requires that you correctly define the Emulator and ROM/Game names and match the control Input Codes to the emulator's configuration. Each emulator passes (to LEDBlinky) the emulator name and game name when you launch a game. These values must be defined correctly – use the [Import Unknown Games](#) option. Using the emulator and game names, LEDBlinky will then match the controls you have defined (using the Controls Editor). But that's only half the job – to correctly light up the LEDs, the Input Codes (keycodes) for each control must match those used by the emulator. For example; if Player 1, Button 1 is configured by the emulator to use keyboard button "A", you must assign P1_BUTTON1 to KEYCODE_A. Detailed instructions are provided below.

Menus



Save

Save the controls configurations. The LEDBlinky controls configuration file is *LEDBlinkyControls.xml* located in the LEDBlinky folder. The “Save” menu option is only enabled when one or more controls have been added or modified. If you attempt to close the Controls Editor app prior to saving, you will be prompted to save the data.

Note: Do not edit the *LEDBlinkyControls.xml* file manually.

Copy ROM/Game

Copy all controls for any ROM/Game to a new ROM/Game name. The new set of controls can be created for any emulator. You must select a specific ROM/Game to enable the “Copy ROM/Game” menu option. After you click the menu option, you will be presented with a dialog to select the destination Emulator.

Copy Player 1 Controls

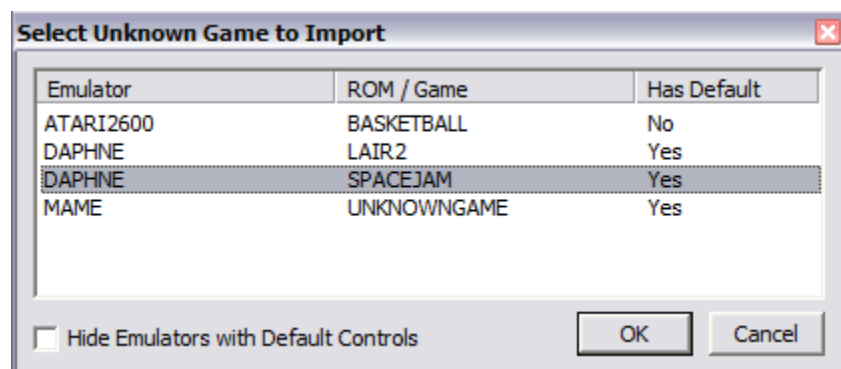
Copy the current set of Player 1 Controls to all other players for the selected game. Most multiplayer games use the same controls for each player – this feature allows you to define the controls for player 1 and then quickly copy them to all other players.

Note: Input codes are never copied across players – they must be uniquely defined for each player control.

Note: Player 1 controls will only be copied to other players for which no controls have been defined.

Import Unknown Games

When configuring new Emulators or Games, start with this option! Each time you play an unknown game, LEDBlinky stores the Emulator name and ROM/Game name as specified by your front-end. Click the “Import Unknown Games” menu option to display the unknown games list.



Select the ROM/Game you wish to import and click “OK”. The controls editor lower pane will open with the game name pre-populated. Do not change ROM/Game Name. You can then proceed to add all the necessary controls.

Note: If the emulator does not exist for the selected game, it will be automatically added. You may wish to edit the Emulator Description.

If no default controls have been specified for the selected Emulator, you will be given the option to import the selected ROM/Game or create a default control group for the emulator. The default control group will be designated as <default>. A default control group is very useful when most or all games for a specific emulator use the same controls. The default controls are only used by LEDBlinky when no individual control group has been defined for the ROM/Game.

Note: You can add a Default Control Group at any time – but only one is allowed for each emulator.

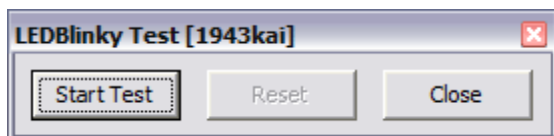
Import MAME Games

The controls for most MAME games are predefined by the *Controls.ini*, *Colors.ini*, and *Mame.xml* files. If you wish to override the predefined control values, use the “Import MAME Games” menu option. You must know the exact ROM name to import the MAME game. After importing the game, you may edit any or all of the controls.

Note: Once a MAME game has been imported, LEDBlinky will not use any control values for that game from the *Controls.ini*, *Colors.ini*, and *Mame.xml* files. If you wish to revert back to the predefined control values, delete the game from the ROM/Game list.

LEDBlinky

Use the “LEDBlinky” menu option to test the configured game start options for any game. All Speech and LED options will activate exactly as if the game was started from your front-end. The “LEDBlinky” menu option is only enabled when a ROM/Game is selected.



Click “Start Test” to simulate running the game. After the speech and LED activity has completed, you can click “Reset” to test again.

Note: All Game Start options are configured using the [LEDBlinky Configuration](#) app.

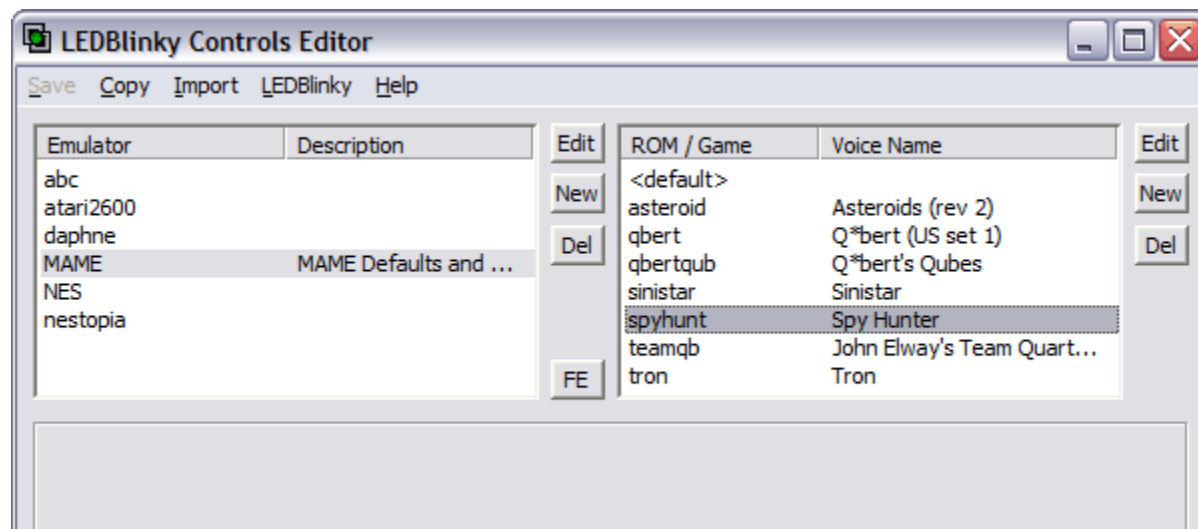
Help

Launches the LEDBlinky help documentation – this document!

Emulator and ROM/Game Lists

The LEDBlinky Controls Editor main window consists of an upper and lower pane. The upper pane contains the Emulator and ROM/Game lists. When the Controls Editor is first opened, the lower pane will be blank.

Note: Emulator and ROM/Game modifications (Edit/New/Delete) are not permanent until you click the “Save” menu option. If you wish to undo modifications, close the LEDBlinky Controls Editor without saving and then reopen the editor.



Emulator List

The left upper pane displays the list of emulators you have configured. Select an emulator in the list to load the list of games configured for that emulator. Double-clicking an emulator in the list will load the emulator into the lower pane for editing. Using the column headers, you can increase or decrease the width of each column.

Note: If you have an emulator designated as the default, it will be listed as “<default>”. See [below](#) for a description of the default emulator.

Edit (Emulator)

Click the “Edit” button to the right of the Emulator List to edit the selected emulator. This will load the emulator into the lower edit pane and disable the upper list panes. The “Edit” button is only enabled when an emulator is selected. See [below](#) for a description of the emulator fields.

Note: The upper list panes will remain disabled until the “OK” or “Cancel” button is clicked on the lower edit pane.

New (Emulator)

Click the “New” button to the right of the Emulator List to add a new emulator. This will display a blank emulator in the lower edit pane and disable the upper list panes. See [below](#) for a description of the emulator fields.

Note: Rather than manually adding a new emulator name, it is highly recommended that you use the [Import Unknown Games](#) option.

Note: The upper list panes will remain disabled until the “OK” or “Cancel” button is clicked on the lower edit pane.

Delete (Emulator)

Click the “Del(ete)” button to the right of the Emulator List to delete the selected emulator. You will be asked to confirm the delete action. The “Del” button is only enabled when an emulator is selected.

Note: Deleting an emulator will also delete all ROM/Games listed for that emulator.

FE

Click the “FE” button to edit the controls for your front-end. FE controls are similar to emulator controls. See [below](#) for a description of the FE controls.

Note: Use the LEDBlinky Configuration app to specify your front-end.

ROM / Game List

The right upper pane displays the list of games you have configured for the currently selected emulator. Select a ROM/Game in the list to enable the edit buttons. Double-clicking a ROM/Game in the list will load the game into the lower pane for editing. Using the column headers, you can increase or decrease the width of each column.

Note: If you have a game designated as the default for the selected emulator, it will be listed in the ROM/Game list as “<default>”. See [below](#) for a description of the default game.

Edit (ROM/Game)

Click the “Edit” button to the right of the ROM/Game List to edit the selected game. This will load the game into the lower edit pane and disable the upper list panes. The “Edit” button is only enabled when a game is selected. See [below](#) for a description of the game fields.

Note: The upper list panes will remain disabled until the “OK” or “Cancel” button is clicked on the lower edit pane.

New (ROM/Game)

Click the “New” button to the right of the ROM/Game List to add a new game. This will display a blank game in the lower edit pane and disable the upper list panes. See [below](#) for a description of the game fields.

Note: Rather than manually adding a new ROM/Game name, it is highly recommended that you use the [Import Unknown Games](#) option.

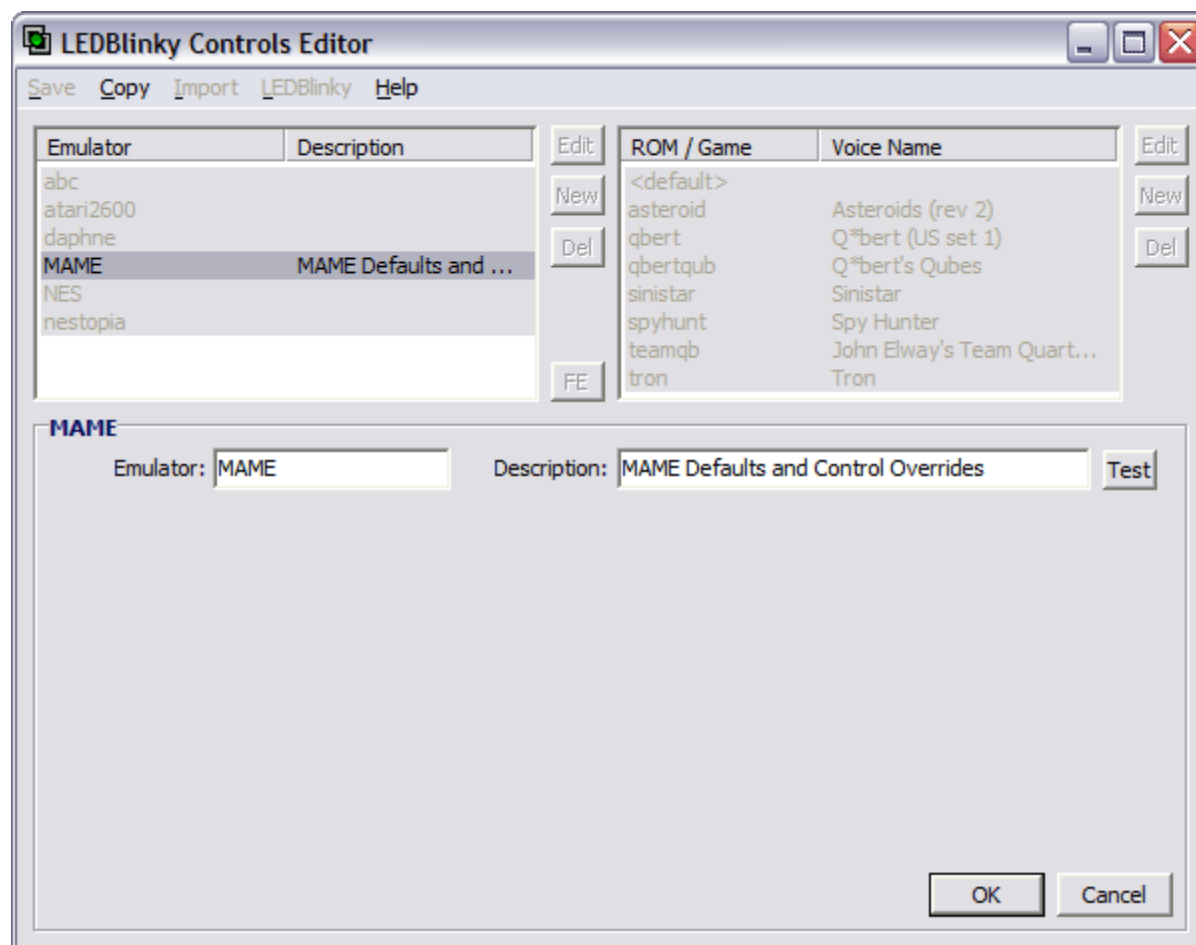
Note: The upper list panes will remain disabled until the “OK” or “Cancel” button is clicked on the lower edit pane.

Delete (ROM/Game)

Click the “Del(ete)” button to the right of the ROM/Game List to delete the selected game. You will be asked to confirm the delete action. The “Del” button is only enabled when a game is selected.

Edit Emulator

The LEDBlinky Controls Editor main window consists of an upper and lower pane. The lower “edit” pane is displayed when you add a new emulator or edit an existing emulator. While editing an emulator, the upper list panes will be disabled.



Emulator

Enter or modify the Emulator name. This value **MUST** match exactly the name passed by the front-end (to LEDBlinky). The only exception to this rule is for MAME – the name “MAME” will apply to any version of MAME. If you have an emulator checked as the default, the name will be listed as “<default>” and cannot be edited. If an emulator name has been loaded using the Import Unknown Games option, do not change this value.

Note: The Emulator name cannot include any spaces - spaces will automatically be replaced with underscores (_).

Note: Rather than manually adding a new emulator name, it is **highly recommended** that you use the [Import Unknown Games](#) option.

Description

Enter any description for the emulator. Currently this field is only used for reference. Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) options – which is pointless because this text is not currently used by LEDBlinky 😊.

Default

Default emulator controls apply to any game for which no other game specific or emulator specific controls have been defined – these are global controls for all emulators (accept MAME).

The “Default” check box is only available when you are creating a “New” emulator and no other emulator has been designated as the default – you can have only one default emulator. Checking the “Default” option will set the Emulator name to “<default>”. You cannot alter the default emulator name.

Note: It is not necessary to create default (global) emulator controls – it is more accurate to define a set of [default](#) controls for each specific emulator.

OK

Click the “OK” button to accept the emulator modifications. Changes or additions will be reflected in the upper left list. After clicking “OK”, the lower edit pane will be blank and the upper list panes will be re-enabled.

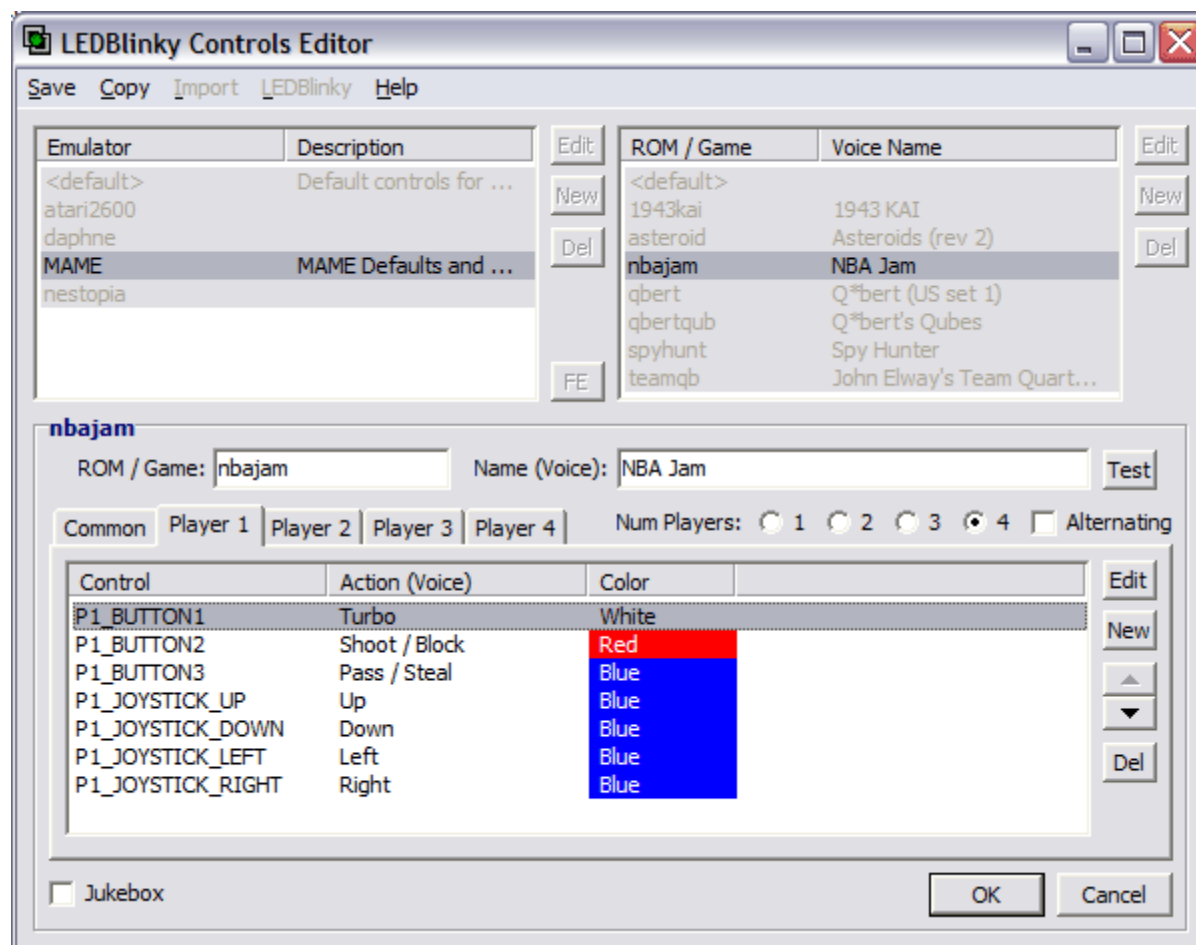
Note: Modifications are not saved to the *LEDBlinkyControls.xml* file until the “Save” menu option is clicked.

Cancel

Click “Cancel” to abandon all emulator modifications. After clicking “Cancel”, the lower edit pane will be blank and the upper list panes will be re-enabled. You can also use the “ESC” key as a shortcut.

Edit ROM/Game and Controls

The LEDBlinky Controls Editor main window consists of an upper and lower pane. The lower “edit” pane is displayed when you add a new game or edit an existing game. While editing a game, the upper list panes will be disabled.



ROM / Game

Enter or modify the ROM/Game name. This value **MUST** match exactly the name passed by the front-end (to LEDBlinky). If you have a game checked as the default, the name will be listed as “<default>” and cannot be edited. If a game name has been loaded using the Import Unknown Games option, do not change this value.

Note: The ROM/Game name cannot include any spaces - spaces will automatically be replaced with underscores (_).

Note: Rather than manually adding a new game name, it is **highly recommended** that you use the [Import Unknown Games](#) option.

Name (Voice)

Enter the game name. The name text will be used to speak the game name when the [option](#) is enabled. Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) options.

Note: You may need to alter the spelling to have the name sound phonetically correct.

Default

Default ROM/Game controls apply to any game for which no other game specific controls have been defined for the current emulator. The default control group also allows you to define controls which are “Always Active” – these controls will be included with every game under this emulator even when a game has its own set of controls defined.

The “Default” check box is only available when you are creating a “New” emulator and no other emulator has been designated as the default – you can have only one default emulator. Checking the “Default” option will set the Emulator name to “<default>”. You cannot alter the default emulator name.

Common Tab

All games have a “Common” player tab which lists all controls that are common across players – usually administration buttons or the [Primary Control](#) type. The “Common” tab also includes the “Default Active” and “Default Inactive” control values.

Player 1/2/3/4 Tabs

Use the “Player Tabs” to display the Controls List defined for each player. The number of player tabs is determined by the “Num Player” option (see below).

Num Players

Select the number of players for the game. Changing the number of players will add or remove “Player Tabs”.

Alternating

Check the “Alternating” option if a game uses one set of controls and alternates the players.

Note: Checking the “Alternating” option will change the “Player Tabs” to only display Common and Player 1 regardless of the “Num Players” setting.

Controls List

The Controls List displays the controls for the selected player (or Common). Select a control in the list to enable the edit buttons. Double-clicking a control in the list will display the [Control Edit](#) window. Using the column headers, you can increase or decrease the width of each column.

Note: If you have a control designated as the [Primary Control](#), the control name will be prefixed with <pri>.

Default Active/Inactive Controls

The first two controls in the “Common” controls list will be the “Default_Active” and “Default_Inactive” controls. These are not true controls – they are only used to define color or intensity. The “Default_Active” color/intensity is used for any active control (used by the game) that does not have a specific color/intensity defined. The “Default_Inactive” color/intensity can be used to light up any non-active control (not used by the game).

Note: The “Default Active/Inactive” controls cannot be deleted or reordered in the Control List, and they are never spoken.

Edit (Control)

Click the “Edit” button to the right of the Control List to edit the selected control. This will load and display the Control Edit window. The “Edit” button is only enabled when a control is selected. See [below](#) for a description of the control fields.

New (Control)

Click the “New” button to the right of the Control List to add a new control. This will display the Control Edit window. See [below](#) for a description of the control fields.

Move Control Up/Down

Use the Up/Down “Arrow” buttons to move the selected control up or down in the Controls List. The Arrow buttons are only enabled when a control is selected.

Note: The order of the Control List determines the order in which the control actions will be spoken before the game starts when the [option](#) is enabled.

Delete (Control)

Click the “Del(ete)” button to the right of the Control List to delete the selected control. You will be asked to confirm the delete action. The “Del” button is only enabled when a control is selected.

Jukebox

Check to designate the game as a jukebox application. This option is used in conjunction with the [Game Play Animation For Jukebox Only option](#).

Note: Any ROM/Game control group (including an emulator default control group) can be designated as a jukebox application. For example, let’s assume you are using an audio animation for Game Play and the Game Play Animation For Jukebox Only option is checked. Using the Controls Editor you have designated your jukebox application. If you also have another game (not a jukebox) and you want the audio animation to run for that game, then just designate it as a jukebox!

Global Pulse

The “Global Pulse” speed is only used for LED-Wiz blinking effects (intensities 129 to 132).

OK

Click the “OK” button to accept the ROM/Game modifications. Changes or additions will be reflected in the upper right list. After clicking “OK”, the lower edit pane will be blank and the upper list panes will be re-enabled.

Note: Modifications are not saved to the *LEDBlinkyControls.xml* file until the “Save” menu option is clicked.

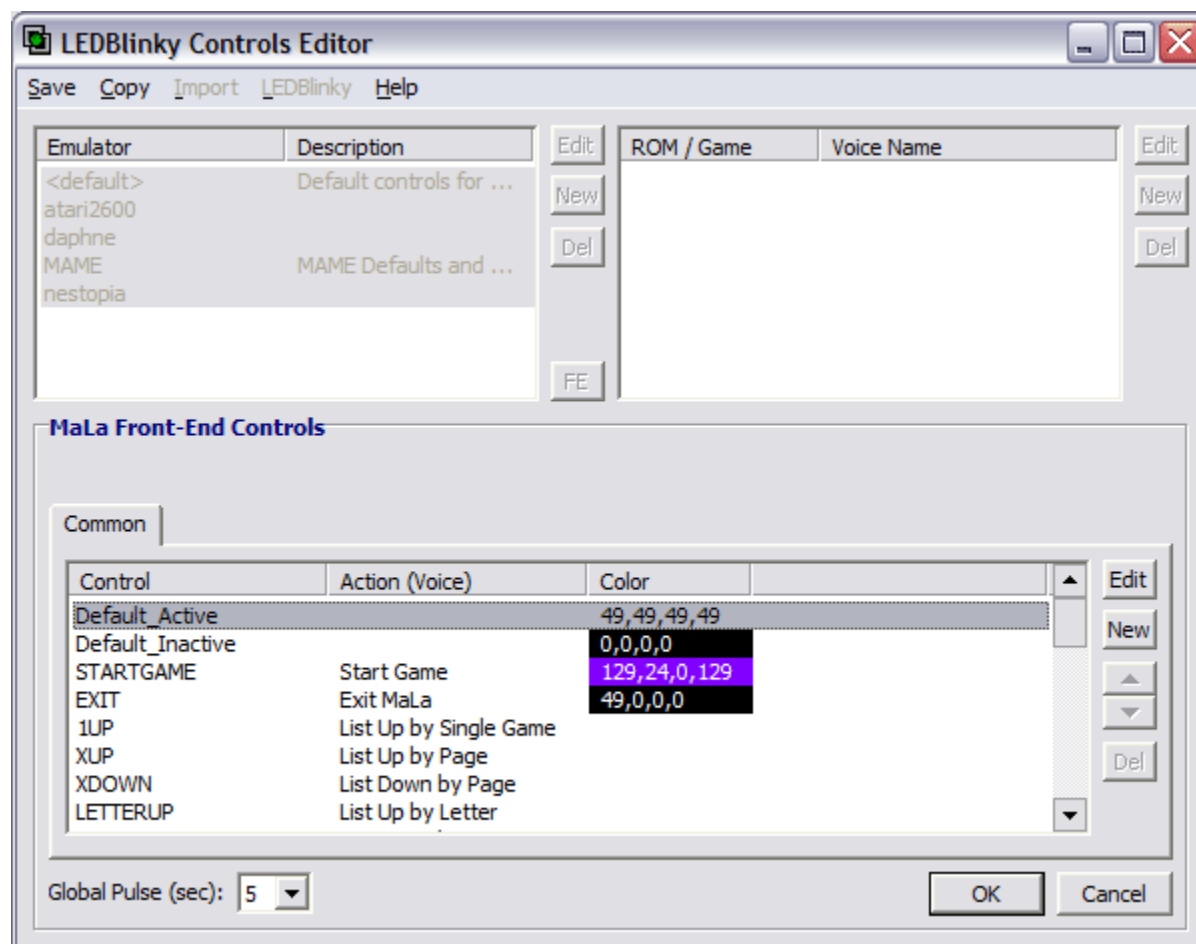
Cancel

Click “Cancel” to abandon all ROM/Game modifications. After clicking “Cancel”, the lower edit pane will be blank and the upper list panes will be re-enabled. You can also use the “ESC” key as a shortcut.

Edit Front-end (FE) and Controls

LEDBlinky can light up the controls for MaLa, AtomicFE, GameEx, HyperSpin, and Maximus Arcade. Select your front-end from the Configuration app FE menu.

The LEDBlinky Controls Editor main window consists of an upper and lower pane. The lower “edit” pane is displayed when you edit the Front-end controls (click the “FE” button). While editing the FE, the upper list panes will be disabled.



Common Tab

Front-end controls are all listed under the “Common” player tab.

Controls List

The Controls List displays the controls for the front-end. Select a control in the list to enable the edit buttons. Double-clicking a control in the list will display the [Control Edit](#) window. Using the column headers, you can increase or decrease the width of each column.

By default, LEDBlinky will include the known controls for your selected front-end. If you do not wish to light up a specific control, you may delete it from the list or just remove the color/intensity for that control.

Edit / New / Move / Delete

These buttons work the same as for the ROM/Game controls. See [above](#).

OK

Click the “OK” button to accept the front-end control modifications. After clicking “OK”, the lower edit pane will be blank and the upper list panes will be re-enabled.

Note: Modifications are not saved to the *LEDBlinkyControls.xml* file until the “Save” menu option is clicked.

Cancel

Click “Cancel” to abandon all front-end control modifications. After clicking “Cancel”, the lower edit pane will be blank and the upper list panes will be re-enabled. You can also use the “ESC” key as a shortcut.

Edit Individual Control

New Player 1 Control for daphne <default>

Control: ☐ Primary Control ☐ Always Active

Name (Voice):

RGB-Color and Single-Intensity

Named Color: Single LED:

RGB Red:

RGB Green:

RGB Blue:

Test Single LED:

Test RGB LED:

Input Codes

Available Input Codes ☐ CP Wired Only

KEYCODE_LCONTROL
JOYCODE_1_BUTTON1
MOUSECODE_1_BUTTON1

< > >>

JOYCODE_1_BUTTON1
JOYCODE_1_BUTTON2
JOYCODE_1_BUTTON3
JOYCODE_1_BUTTON4
JOYCODE_1_BUTTON5
JOYCODE_1_BUTTON6
JOYCODE_1_BUTTON7
JOYCODE_1_BUTTON8

Control

When adding a new control, select the “Control” name from the drop-down list. The list will only include controls that are available for the current player (1, 2, 3, 4, or Common).

Note: You should only add the same control more than once to specify a different action for combo-buttons. For example, if Button 1 and Button 2 can be pressed together for a special action, define each button twice; once for their unique action and once with the same action for the combo.

Primary Control

Check the “Primary Control” option to set the current control as the primary. Only one control for a game can be set as the primary. The “Name (Voice)” for the primary control is [spoken](#) separately from the rest of the controls (and it is not blinked).

The “Primary Control” is also used to determine the [Joystick Digital Restriction](#) mode for u360 and 49-Way joysticks.

Note: See [here](#) for a description on how to override the JDR mode.

Always Active

Check the “Always Active” option to insure that the control will always light up regardless if it is required by the game you are playing. This is useful for lighting administration buttons. The “Always Active” option is only available for the “Default” controls group.

Name (Voice)

The “Name (Voice)” text will be used to speak the control name or action when the [option](#) is enabled. Click the “Test” button to hear how the text will sound using the current [Text To Speech](#) options.

Note: You may need to alter the spelling to have the name sound phonetically correct.

RGB-Color and Single-Intensity

Your control panel may have Single, RGB, or both types of LEDs. These have been defined using the [Generate LEDBlinky Input Map](#) application. If your control panel has only Single type LEDs, the RGB Color options will be disabled. If your control panel has only RGB type LEDs, the Single Intensity options will be disabled.

Note: If your control panel uses both RGB and Single type LEDs, you may want to define both Color and Single values for each control – conceivably you could remap a game function from a RGB LED button to a Single LED button or vice versa.

Named Color

A RGB color can be selected from the “Named Color” list. The individual RGB intensities will be set to match the selected color. The selected “Test RGB LED” will light up with the color.

Note: The “Named Color” option is not available unless you are using LED-Wiz or Pac-LED64 hardware.

Pick Color

Click the “Pick Color” button to select a color from a pre-defined pallet. The individual RGB intensities will be set to match the selected color. If the selected color is also available as a named color, the “Named Color” value will be set. The selected “Test RGB LED” will light up with the color.

Note: RGB LEDs cannot accurately reproduce all the colors you see in the pallet. For example, gray tones do not look correct and black is impossible.

Note: The “Pick Color” option is not available unless you are using LED-Wiz or Pac-LED64 hardware.

RGB Red / Green / Blue

The LED color can be defined by selecting the individual intensity values for the Red, Green, and Blue color components. The selected “Test RGB LED” will light up the specified color.

If your control panel uses LED-Wiz hardware, then the intensity values will be 0 to 48, plus the built-in hardware blinking effects. If your control panel uses Pac-LED64 hardware, then the intensity values will be 0 to 48. If your control panel uses PAC-Drive or U-HID hardware, then the intensity values will be *On* or *Off*.

Note: If you are using any blinking effects for RGB LEDs and your control panel has more than one LED device, all three color leads for each RGB should be physically wired to the same LED device. Wiring the colors across devices will cause out-of-sync blinking effects with very strange results.

Single LED

Select the intensity value for the single color type LED. The selected “Test Single LED” will light up at the specified intensity.

If your control panel uses LED-Wiz hardware, then the intensity values will be 0 to 48, plus the built-in hardware blinking effects. If your control panel uses Pac-LED64 hardware, then the intensity values will be 0 to 48. If your control panel uses PAC-Drive or U-HID hardware, then the intensity values will be *On* or *Off*.

Test Single LED

Select a LED for testing any single intensity from the list of LED Labels as defined in your input map. The same LED on your control panel will be used for testing no matter which control you are setting.

Test RGB LED

Select a LED for testing the RGB intensities from the list of LED Labels as defined in your input map. The same LED on your control panel will be used for RGB testing no matter which control you are setting.

Input Codes

Defining the correct “Input Codes” for a control is critical to getting that control to light up! The “Input Codes” (keycodes) for each control must match those used by the emulator. For example; if Player 1, Button 1 is configured by the emulator to use keyboard button “A”, you must assign KEYCODE_A to the P1_BUTTON1 control.

Note: Defining “Input Codes” is only required for non-MAME emulator controls. MAME and Front-end input codes are pre-defined and the input code fields will be disabled.

Using MAME Default Input Codes Option

If you have checked the “Use MAME Default Control Mapping for Other Emulators” option, then all input code fields will be disabled. You can change this option from the

Configuration app or click the “Question” button. See [here](#) for a full description of this option.

Quick Key

Click this button to quickly select and assign a keyboard input code. When the button is depressed, it will flash ‘Press Key’; then press any key on the keyboard. The input code will be selected and added to the Input Codes list. To abort the Quick Key mode, click the button a second time.

Note: The Quick Key feature cannot be used to assign joystick buttons – these must be selected manually from the list.

Available Input Codes

Select an input code from the list of “Available Input Codes” and click the “<” button to add the value to the Input Codes list for the control. Use the “>” button to remove a single (selected) value or the “>>” button to remove all values from the Input Codes list for the control. Double-clicking an input code in either list will add or remove the value from the Input Codes list.

CP Wired Only

Check the “CP Wired Only” option to filter the “Available Input Codes” list to only input codes that you have assigned (wired) to your control panel. These are the only values you should be assigning to the controls.

OK

Click the “OK” button to accept the control modifications. Changes or additions will be reflected in the Control List.

Note: Modifications are not saved to the *LEDBlinkyControls.xml* file until the “Save” menu option is clicked.

Cancel

Click “Cancel” to abandon the control modifications. You can also use the “ESC” key as a shortcut.

How to Specify JDR Mode/Map Overrides

During normal operation, the “Primary Control” is used to determine the [Joystick Digital Restriction](#) mode for u360 and 49-Way joysticks. Each primary control maps to a JDR mode/map. For example, all games assigned CONTROL_STICK as the primary control will use the u360 *Analog.um* map and/or the GP-Wiz49 Raw49 mode. (stick = analog? I know, it makes no sense). See [here](#) for a complete list of the Primary Control Mode/Map associations.

For the u360, you may wish to create your own maps using the UltraMap software provided by Ultimarc. Your own custom map can be substituted for any map included in the */LEDBlinky/JDR/UltraStik/* folder. For example, if you replace the *joy8way.um* file with your own version, all games that are assigned the CONTROL_JOY8WAY primary control will use your new map.

If you wish to assign a custom u360 map to one or more specific games, then follow these steps:

- 1) Put your new map file in the */LEDBlinky/JDR/UltraStik/* folder. Make sure it has a unique name – do not overwrite an existing map file.
- 2) Edit the LEDBlinkyControls.xml file using Notepad.
- 3) Locate the control group node for the game you wish to modify. For example;
`<controlGroup groupName="tron" numPlayers="2" alternating="1">`
- 4) Under the control node, locate the primary control node. If the primary control node does not exist, you must first add a primary control using the LEDBlinky Controls Editor. For example;
`<control name="CONTROL_JOY8WAY" primaryControl="1" />`
- 5) Add the UltraStik attribute and set equal to the name of your new map file. You do not need to include the “.um” file extension (but you can if you wish). For example;
`<control name="CONTROL_JOY8WAY" primaryControl="1" ultrastik="mymap" />`
- 6) Save and close the file.

File Descriptions

Colors.ini

This file defines game specific control colors (or intensities) for MAME. A version of the file with authentic control panel colors supporting over 1200 ROMs (at last count) has been included with LEDBlinky. An updated version may be available for downloaded from the ArcadeControls forum.

Color-RGB.ini

This file contains a list of Color Names and the RGB (Red/Green/Blue) intensity values for each. Intensity values must be in the range of 0 to 48. Note: some colors cannot be reproduced by a RGB LED, for example; shades of Gray or Black. In these cases, substitute colors must be specified. This file is only used in conjunction with an optional *colors.ini* file for MAME.


GenLEDBlinkyInputMap.exe

Generate LEDBlinky Input Map Application. This app will allow you to generate or update the *DevicePort-Input.ini* file required by the LEDBlinky plug-in, Controls Editor, and Animation Editor. The DevicePort-Input file defines the connection between the LED Output Control (LED-Wiz, Pac-LED64, etc.) device ports, and the Input Codes (Keyboard, Trackball, Joysticks, etc). It also defines the three ports used by each RGB LED (if any). This mapping should never change unless you rewire your LED Output Control device(s) or reprogram your keyboard encoder. See [here](#) for a complete description.

LBkbh.dll

This dll provides LEDBlinky with the ability to abort the Game Start and Game Pause speech features by pressing any key. Keyboard monitoring is only active when LEDBlinky is speaking the Game Start and Game Pause options. No keystrokes are recorded or retained.

LEDBlinky.exe

Core LEDBlinky application. This app runs in the Windows system tray . It accepts commands using [command-line](#) parameters or via a plug-in interface. See [here](#) for a complete description.

LEDBlinky.mplugin

This is the MaLa plug-in (dll).

LEDBlinky.atoplug

LEDBlinky.plugcfg

This is the AtomicFE plug-in and plug-in configuration file.

LEDBlinky_GX.dll

This is the GameEx plug-in (dll).

LEDBlinkyAnimationEditor.exe

LEDBlinky Animation Editor Application. This app will allow you to create or modify LED Animations. The animation editor can load all animation formats (LWA or LWAX), but will only generate LWAX output files. LWAX animation files are not exclusive to the LEDBlinky plug-in; they can be played by any LED application supporting the format. Although the Animation Editor can be used without the LEDBlinky plug-in, it does require a *DevicePort-Input.ini* file which can be generated using the [LEDBlinky Input Map Application](#). See the *Animation Editor.pdf* file for a complete description.

LEDBlinkyConfig.exe

LEDBlinky Configuration Application. This app will allow you to configure the LEDBlinky plug-in. It also provides quick access to the other LEDBlinky tools. See [here](#) for a complete description.

LEDBlinkyControls.xml

This file stores all LEDBlinky control related information – this includes Emulator controls, MAME control overrides, Front-end controls, control defaults, and other stuff. You should never need to edit this file manually and doing so may result in LEDBlinky errors. The *LEDBlinkyControls.xml* file is modified by the [LEDBlinky Controls Editor](#).

LEDBlinkyControlsEditor.exe

LEDBlinky Controls Editor. This app allows you to configure the controls for any Emulator, Game, or Front-end. For each control you can specify the color or intensity, the voice 'action', and if necessary the input codes associated with the control. Controls can be defined as the default for an emulator, or individually for a specific game. See [here](#) for a complete description.

LEDBlinkyTroubleshooter.exe

The LEDBlinky troubleshooting app will attempt to provide solutions for common issues such as the wrong buttons lighting up during game play. The app is self-explanatory and may provide additional information to help resolve your problem.

Ledwiz.dll

LED-Wiz Function Library (Created by MikeQ). Provides the interface to the LED-Wiz device(s).

LibUSB0.dll

Used exclusively by the UltraStik U360 Control library (ultrastik.dll). If you do not have any U360 joysticks, then this file may be deleted along with the /jdr/UltraStik folder.

Mame.dll

MAME Outputs Function Library (Ceated by headkaze © Copyright 2009 Headsoft www.headsoft.com.au). Provides the output interface to MAME (version .112 or later).

MameOutputs.ini

This file maps MAME Outputs to specific controls or device ports. The purpose of each output varies from game to game, but LED0 and LED1 can generally be assumed to flash the player start buttons when credits are available.

Any output can be added to the *MameOutputs.ini* file. To define outputs for a specific game, create an output section with the section key equal to the rom. For example, [digdug]. The [default] outputs will be used for any game which does not have its own output section. See the *MameOutputs.ini* file for a full description of the output key=value format.

Use the MAME Output Test app to determine which (if any) outputs are generated by a specific game.

MameOutputTest.exe

This application will list all outputs generated by a MAME game. Start the application, then start MAME (or MAME32), then start any game. Outputs will be listed in real-time as they change state; On/Off.

NewInputCodes.ini

This file is used to add new MAME Input Codes. You should only edit this file when a new release of MAME adds new input codes. Each Input Code must be mapped to an existing LEDBlinky Base Input Code. Base input codes are defined in *LEDBlinkyControls.xml* file under the <baseInputCodes> node.

Pacdrive.dll

Pac-LED64, Pac-Drive / U-HID / ServoStik Function Library (Created by headkaze © Copyright 2009 Headsoft www.headsoft.com.au). Provides the interface to the Pac-LED64, Pac-Drive, U-HID LED controllers, and also ServoStik joysticks.

Settings.ini

This file stores all LEDBlinky configuration settings. Use the LEDBlinky Configuration app to modify this file – do not edit manually.

UltraStik.dll

UltraStik U360 Function Library (Created by headkaze © Copyright 2009 Headsoft www.headsoft.com.au). Provides the interface to U360 joysticks.

ZipDll.dll

Freeware zip library designed by Eric Engler. Available for download here:

<http://www.geocities.com/SiliconValley/Network/2114/>

Also see: <http://www.info-zip.org/>

Credits

I'd like to thank the following persons for their help (directly and indirectly) with the LEDBlinky Arcade LED Control software and Animation Editor.

- Swindus and Loadman – The developers of MaLa; an excellent FE!
- Youki – The developer of AtomicFE; another excellent FE!
- Tom S. – The developer of GameEx; yet another excellent FE!
- BadBoyBill. – The developer of HyperSpin; yet another excellent FE!
- Loadman – The pioneer of many MaLa plug-ins. Thanks for all your support and for not getting pissed-off when I decided to release a “similar” plug-in.
- Headkaze – Thanks for providing the colors.ini file – with that data, the LEDBlinky can light up the buttons in their original arcade colors! Also, thanks for the PacDrive dll, Mame dll, UltraStik dll, and other code support.
- Kevin Jonas (SirPoonga) and Howard Casto – Thanks for providing Controls.dat. Without that data, many of LEDBlinky's cool features would not be possible.
- And many thanks to all the other ArcadeControls Forum members who have suggested new LEDBlinky features!

Index

Abort Speech, 19
All Available Input Codes, 44
Alternating, 55
Always Active, 61
Amp Mult, 31
Amplitude Animation, 29
Audio Animation Mode, 28
Audio Device, 28
Audio Input, 28
Available Input Codes, 63
Berzerk Mode, 22
Blink Controls, 29
Button Flash, 20
Cabinet LEDs, 34
Cabinet LEDs Animation, 28
Cancel, 53, 57, 59, 63
Clear Port, 44
Cocktail Table Mode, 38
Color/Intensity Spectrum, 28
Color-RGB.ini, 65
Colors.ini, 25, 65
Common Tab, 55, 58
Control, 60
Control Drop-down List, 30
Control Panel Max Players, 37
Controller File, 25
Controls Editor, 13
Controls List, 55, 58
Controls.ini, 26
Copy Player 1 Controls, 47
Copy ROM/Game, 47
CP Wired Only, 63
Dead Zone, 32
Decay, 32
Default, 53, 55
Default Active/Inactive Controls, 56
Delete (Control), 56
Delete (Emulator), 50
Delete (ROM/Game), 51
Demo Duration, 24
Demo Game Controls, 24
Demo Scroll Delay, 24
Description, 53
Device, 43
Display Decay, 31
Display Errors On FE Quit, 39
Display Peaks, 32
Edit (Control), 56
Edit (Emulator), 49
Edit (ROM/Game), 50
Edit / New / Move / Delete, 59
Emulator, 52
Emulator List, 49
Enable Debug Log, 38
Enable JDR Mode, 36
Exclude Selected Animations From
 Random, 38
FE, 13, 50
FE Active Animation, 21, 28, 35, 36
FE List Change Animation, 21, 36
FE Quit Animation, 21
FE Start-up Animation, 20, 35, 36
Flash Player Start with Credits + Other
 MAME Outputs, 17
Game Pause Animation, 16, 28
Game Play Animation, 16, 28
Game Play Animation For Jukebox
 Only, 16
Game Start Animation, 17
Generate Input Map, 13
GenLEDBlinkyInputMap.exe, 65
Global Pulse, 57
Help, 14
HyperSpin Settings.ini, 24
Import MAME Games, 48
Import Unknown Games, 47
Input Codes, 62
Jukebox, 56
LBkbh.dll, 65
LED Animation Editor, 13
LED Device(s), 35
LED Type, 43
LEDBlinky.atoplug, 65
LEDBlinky.exe, 65
LEDBlinky.mplugin, 65
LEDBlinky.plugcfg, 65
LEDBlinky_GX.dll, 65
LEDBlinkyAnimationEditor.exe, 66

LEDBlinkyConfig.exe, 66
LEDBlinkyControls.xml, 66
LEDBlinkyControlsEditor.exe, 66
Ledwiz.dll, 66
LibUSB0.dll, 66
Light AtomicFE Controls, 23
Light Game Controls, 15
Light GameEx Controls, 23
Light HyperSpin Controls, 23
Light MaLa Controls, 23
Light Maximus Arcade Controls, 23
Light Player Start And Coin Buttons, 15
Lock Frequency, 30
Lock Threshold, 31
MAME Config Folder, 25
MAME Output Test, 13
Mame.dll, 66
MAME.xml, 26
MameOutputs.ini, 66
MameOutputTest.exe, 67
Maximus Default.ini, 24
Move Control Up/Down, 56
Name (Voice), 55, 61
Named Color, 61
New (Control), 56
New (Emulator), 49
New (ROM/Game), 50
NewInputCodes.ini, 67
No LEDs Mode, 38
Num Players, 55
OK, 53, 57, 59, 63
Pacdrive.dll, 67
Pick Color, 61
Player 1/2/3/4 Tabs, 55
Port, 43
Port Input Codes, 43
Port Label, 43
Preload Audio Library, 38
Primary Control, 60
Pulse Animation, 28
Quick Key, 44, 63
Random Text Delay, 22
Restrictor Plate Installed, 37
RGB Red / Green / Blue, 61
RGB-Color and Single-Intensity, 61
ROM / Game, 54
ROM / Game List, 50
Run Animation When FE is Active, 35
Run Animation When Game is Active, 35
Sample Rate (ms), 32
Save, 13, 45, 47
Screen Saver Animation, 22, 28
Set Port, 44
Settings.ini, 67
Single Amp Range, 30
Single LED, 62
Speak And Blink AtomicFE Controls, 24
Speak and Blink Controls on Game
 Pause, 17
Speak and Blink Controls On Game
 Start, 18
Speak And Blink GameEx Controls, 24
Speak And Blink HyperSpin Controls, 24
Speak And Blink MaLa Controls, 24
Speak And Blink Maximus Controls, 24
Speak Name On Game Start, 18
Speak Primary Controls On Game Start,
 18
Speak Random Text, 22
Speak Text On FE Quit, 21
Speak Text On FE Start, 20
Speak Text On Game Start, 18
Spectrum Colors, 29
Spectrum Mode – Color Fade, 29
Spectrum Mode – Color Transition, 29
Speech Rate, 34
Speech Voice, 34
Speech Volume, 34
Spoken Controls Prefix, 18
Strobe LEDs with Speech, 18, 21
Test, 28
Test Single LED, 62
Tools, 13
Trigger Frequency, 32
Trigger Mode – Increase Amplitude, 30
Trigger Mode – Trigger Threshold, 30
Trigger Threshold, 31
UltraStik 360, 37
UltraStik.dll, 67
Use MAME Default Control Mapping for
 Other Emulators, 26

Use MAME to Trigger the Game
Start/Stop Events, 26
Using MAME Default Input Codes
Option, 62

Verbose Logging, 38
ZipDll.dll, 67