

Raspberry Pi Setup Guide

From Sinden Lightgun

Contents

- 1 Introduction
 - 1.1 Recommended Image
 - 1.2 DIY Recommendation
 - 1.3 Game Compatibility
- 2 Video Setup Guides
 - 2.1 BareBones Basic v7 (by Titchgamer)
 - 2.2 Barebones v8 (by Wiggy808)
 - 2.3 2 Player Troubleshooting
- 3 Sinden Lightgun setup on RetroPie
 - 3.1 Preface
 - 3.2 Things you will need
 - 3.3 Downloading the drivers
 - 3.4 Setting up RetroPie, lightgun drivers and basic calibration
 - 3.4.1 Preparing the SD Card
 - 3.4.2 Configuring the Raspberry Pi
 - 3.4.3 Copying & Installing the Sinden software
 - 3.4.4 Patching/Updating RetroArch
 - 3.5 Creating a basic mapping for the lightgun
 - 3.6 How to map individual games
 - 3.6.1 Border overlay and override files
 - 3.6.2 Lightgun mappings
 - 3.7 Some convenience and pretty stuff
 - 3.7.1 Pretty entries for Lightgun utilities in EmulationStation
 - 3.7.2 Run Lightgun drivers automatically
 - 3.8 Patched lr-mame2016
 - 3.9 In closing
 - 3.10 DC/Naomi/Atomiswave

Introduction

To get started with Sinden setup on RPi it is recommended you use a fresh, Lightgun dedicated image.

Recommended Image

BareBones is the recommended image. Created by HarryDog and updated by the community it comes with all emulators and additional configuration files setup. Just add your own games and bios files.

You can download it from the Lightgun.tech Website:

Lightgun.tech Website (<https://lightgun.tech/>)

DIY Recommendation

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)

[OK](#)

Game Compatibility

Compatibility List (<https://lightgun.tech/game-compatibility-list/>)

Video Setup Guides

BareBones Basic v7 (by Titchgamer)

How to setup a Raspberry Pi 4 for use with the Sinden Ligh...



Barebones v8 (by Wiggy808)

Sinden Lightguns BareBones v8 Pi4 Image install, setup an...



2 Player Troubleshooting

For 2 player issues check out Wiggy808's video for pointers:

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)

[OK](#)

Sinden Lightgun Barebones v7 :: MAME setup for your Ras...



Sinden Lightgun setup on RetroPie

Preface

This guide is intended for people who want to set up the Sinden gun(s) in a “mixed” mode, having both lightgun and non-lightgun games on the device. It is also intended to preserve the bezels from e.g. The Bezel Project (<https://github.com/thebezelproject/>) for non-lightgun games, so we’ll take some extra steps to only add the white borders etc. to the games that need it.

The instructions here should get you 100% of the way in most cases, and 90% of the way in the rest, at least for the more common and well supported emulators, i.e. `lr-flycast`, `lr-fbneo`, `lr-mame2003-plus` (with slight tinkering in the RetroArch’s Controlshere and there), `lr-pcsx-rearmed` and `lr-duckstation`).

It is highly recommended to learn a bit about RetroArch and how it handles controls (<https://docs.libretro.com/guides/input-and-controls/>) first and take a look at how RetroArch configurations work (<https://docs.libretro.com/guides/overrides/#logic>), as we’ll be editing those quite a bit.

Important hint

If you have trouble with a particular game, also make sure to check out the Barebones compatibility list (<https://barebones.lightgun.dev/game-compatibility-list>). Even though you might not be using that image, it uses the same basic system, so any “go/no-go” and notes will apply to your setup in many cases, too. And some of the hints there are great!

ATTENTION: If you’ve already got a working RetroPie setup or a pre-made image running, feel free to skip to “Copying & Installing the Sinden software”.

Things you will need

- A Raspberry Pi (<https://www.raspberrypi.org/>) - This is a credit card sized single board computer. In this tutorial I will be using a Raspberry Pi 4 with 4GB of ram.
- A micro SD card - This is a small memory card that will be used for storage.

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)[OK](#)

- A Sinden Light Gun (or two) - Although it is possible to set up the system without the gun, you will need one to test it.
- A game controller (optional but recommended) - RetroPie is designed to be navigated with a game pad. You can configure it to be navigated with keyboard controls, but it is not as intuitive.
- A PC running Windows with an SD card reader - There are ways of setting the pi up without this, but to make things as simple as possible you'll need a PC.

Downloading the drivers

First, download the the latest Sinden beta version (<https://www.sindenlightgun.com/software/PiImprovedResilienceBeta.zip>)

Setting up RetroPie, lightgun drivers and basic calibration

Preparing the SD Card

Now after downloading the latest RetroPie image for your pi model from the RetroPie page (<https://retropie.org.uk/>), it is time to set it up. Extract the file, and use Win32DiskImager (<https://win32diskimager.download/download-win32-disk-imager/>) or Raspberry Pi Imager (<https://www.raspberrypi.org/software/>) (or some other software) to write the .img file to your SD card. Once the process is complete, remove your SD card from your windows PC and pop it in your Pi. Note that if you flash RetroPie as a "Custom Image" in Raspberry Pi Imager, after selecting the image and the target drive, you can use CTRL+SHIFT+X to open a hidden menu where you can turn on SSH and your configure your hostame and wifi connection.

Configuring the Raspberry Pi

Connect power and your display, game controller, keyboard, and power up your Pi.

The first screen you see after loading will tell you a gamepad is connected, and you need to configure it. Follow the instructions on screen. Remember your hotkey, I always set it to the central XBox button on my controller, or the "coin" button on arcade cabinets.

The next step is to connect your pi to the internet. You can do this by setting up the inbuilt Wi-Fi on the pi, or you can just pop a network cable in the Pi and the other end into your router.

Then we need to give the Windows PC remote access to the Pi:

- Select Retropie from the main Emulation Station screen (it's the only option) and then select `raspi-config`
- Select option 3 Interface options
- Enable option 2 Remote command line access using SSH and exit the menu.

In this config tool, you can also configure your wifi connection, for easier access later.

Once that is done return to your Windows PC and install WinSCP (<https://winscp.net/eng/index.php>). This is a file transfer software that will let us drag and drop files from our Windows PC to the Pi (if they are on the same network).

Copying & Installing the Sinden software

Make sure you have the latest beta Linux Sinden software (<https://www.sindenlightgun.com/software/PiImprovedResilienceBeta.zip>) at hand and extract it.

Load WinSCP and start a new session:

- Hostname `retropie`

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

More information

OK

You might get a security warning as this is the first time you connect. Accept the information to continue. The screen should now show your PC's file system on the left and the Pi's on the right. Copy the folder called **Lightgun** from the Sinden beta folder to the `/home/pi/` directory on the Pi.

Return to the RetroPie. Strangely, we need to install a game before we do anything else. Let's use Tyrian. This is needed to create a `ports` directory on the Pi. Go to **RetroPie setup** in the main RetroPie menu, navigate to **Manage packages** -> **Manage optional packages** -> **opentyrian**. Install the game, exit, and restart the Pi.

You should now have two options in the main menu: **RetroPie** and **Ports**. On the main **RetroPie** screen press **F4** on your keyboard. This will take you to a command prompt. Press **<Enter>** after each command below (please type exactly as given here, including capitalization). We will set execution permissions on the scripts and run them.

```
cd Lightgun
chmod +x setup-lightgun.sh
./setup-lightgun.sh
```

This might take a couple of seconds/minutes.

Then do the same with the `retropie-setup.sh` file:

```
chmod +x setup-retropie.sh
./setup-retropie.sh
```

And finally:

```
emulationstation
```

Now you should be back at your nice graphical interface.

Go into **Ports** and start **SINDENLIGHTGUNTESTP1**. Follow the on-screen instructions to calibrate your gun for Player 1.

Do the same for **SINDENLIGHTGUNTESTP2**, if you have a second gun.

Manual Calibration Note: If you want tight control of the calibration, you can also adapt it manually. You'll need to enter offset values for that: For example if you look down sight and the crosshair is to the left of it, you need to increase your x axis value. You can do this by loading up WinSCP again and editing the `/home/pi/Lightgun/Player1/LightgunMono.exe.config` for Player 1 and `/home/pi/Lightgun/Player2/LightgunMono.exe.config` for Player 2.

Just double click on the file and you can edit it within WinSCP. The lines you need to edit are:

```
<add key="CalibrateX" value="" />
<add key="CalibrateY" value="" />
```

Pop a number between the quotes, save, restart the Pi, start **SINDENLIGHTGUNTESTP1** and launch a game to see how much the sight is off, repeat until you are happy with the results. There's no quick way of doing this I've found.

Player 1 and Player 2 order: If the guns are in the wrong order (e.g. you want a red for P1 and a blue gun for P2, but they show up the other way around), just switch the USB plugs and their ports around or to see the boot order for the Pi4 see this video (<https://youtu.be/or94oTj1Kyw?t=264>).

Before we start gaming, we still need to do a couple of more things. And that is updating/patching retroarch, as the stock version that comes with RetroPie has issues with Lightgun support.

First, we need to start `sudo /home/pi/RetroPie-Setup/retropie_setup.sh` once more, and select **Update RetroPie-Setup script**. Exit RetroPie-Setup after the update completes.

Now, to install the patched version of RetroArch 1.9.9, you can use the `diff` and `sh` files provided by Discord User [@LutherGond#7287](#). I packaged them into a ZIP `cmituRetroArchPatch.zip`. They are based on cmitu's github fork of Retroarch (<https://github.com/cmitu/RetroArch/>) and can be downloaded here (<https://www.mediafire.com/file/um63nzpaq76a5h2/cmituRetroArchPatch.zip/file>).

Download and unzip the file. Copy the file `retroarch.sh` into `/home/pi/RetroPie-Setup/scriptmodules/emulators`, and run `chmod +x retroarch.sh` on it. Copy the other files into the `retroarch` subfolder in that same directory. Overwrite any files that may be already present and add any files that are not (specifically, `04_lightgun_workaround.diff` is new and contains the main lightgun workaround).

Then, start `retropie_setup` (run `sudo /home/pi/RetroPie-Setup/retropie_setup.sh`), go to **Manage packages** -> **core** -> **retroarch**. Remove it and then **Install from source**.

This will take a while, as you're now building and installing RetroArch from scratch, while applying cmitu's patches.

Alright, now you have a version of RetroArch that can handle lightguns properly.

Creating a basic mapping for the lightgun

While working for multiple "mouse"-based devices (such as our lightguns), it seems that retroarch (and maybe other programs, too) don't differentiate between keypresses from multiple keyboards. To get the guns to send non-conflicting keypresses, we need to reconfigure some of the buttons on it.

Do this by editing the mapping `/home/pi/Lightgun/Player1/LightgunMono.exe.config`.

Unfortunately, it seems the keycodes in the PDF that comes with the lightgun drivers ([SindenLightgunDocumentationV1.05_Linux.pdf](#)) is out of date. So you may use my mapping to modify the respective mapping lines to read like this (the ... indicates that there is text between those two blocks to edit that you don't need to touch):

```
<add key="ButtonTrigger" value="1" />
<add key="ButtonPumpAction" value="3" />
<add key="ButtonFrontLeft" value="51" />
<add key="ButtonRearLeft" value="2" />
<add key="ButtonFrontRight" value="49" />
<add key="ButtonRearRight" value="50" />
<add key="ButtonUp" value="66" />
<add key="ButtonDown" value="62" />
<add key="ButtonLeft" value="44" />
<add key="ButtonRight" value="47" />
...
<add key="ButtonTriggerOffscreen" value="3" />
<add key="ButtonPumpActionOffscreen" value="3" />
<add key="ButtonFrontLeftOffscreen" value="51" />
<add key="ButtonRearLeftOffscreen" value="2" />
<add key="ButtonFrontRightOffscreen" value="49" />
<add key="ButtonRearRightOffscreen" value="50" />
<add key="ButtonUpOffscreen" value="66" />
<add key="ButtonDownOffscreen" value="62" />
<add key="ButtonLeftOffscreen" value="44" />
<add key="ButtonRightOffscreen" value="47" />
```

This will set the d-pad to use the wasd keys and the buttons to use the fgh keys, respectively, for gun 1. Similarly, modify `/home/pi/Lightgun/Player2/LightgunMono2.exe.config` to read like this:

```
<add key="ButtonTrigger" value="1" />
<add key="ButtonPumpAction" value="3" />
<add key="ButtonFrontLeft" value="3" />
<add key="ButtonRearLeft" value="3" />
```

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)

[OK](#)

```

<add key="ButtonDown" value="75" />
<add key="ButtonLeft" value="76" />
<add key="ButtonRight" value="77" />
...
<add key="ButtonTriggerOffscreen" value="3" />
<add key="ButtonPumpActionOffscreen" value="3" />
<add key="ButtonFrontLeftOffscreen" value="3" />
<add key="ButtonRearLeftOffscreen" value="2" />
<add key="ButtonFrontRightOffscreen" value="13" />
<add key="ButtonRearRightOffscreen" value="7" />
<add key="ButtonUpOffscreen" value="74" />
<add key="ButtonDownOffscreen" value="75" />
<add key="ButtonLeftOffscreen" value="76" />
<add key="ButtonRightOffscreen" value="77" />

```

This will set the d-pad to use the up/down/left/right keys and the buttons to use the 123 keys, respectively, for gun 2.

In order to have a clean central place to configure the lightgun inputs for RetroArch, I created a file `lightgun.cfg` in `/opt/retroPie/configs/all/retroarch/config/`, with this content:

```

# Never save-on-exit after an override config
# or the override will make into the core config.
config_save_on_exit = false

input_player1_mouse_index = "0"
input_player1_gun_trigger_mbtn = "1"
input_player1_gun_offscreen_shot_mbtn = "2"
input_player1_gun_start_btn = "6"
input_player1_gun_select_btn = "7"
input_player1_gun_dpad_up = "up"
input_player1_gun_dpad_down = "down"
input_player1_gun_dpad_left = "left"
input_player1_gun_dpad_right = "right"
input_player1_gun_aux_a = "num1"
input_player1_gun_aux_b = "num2"
input_player1_gun_aux_c = "num3"

input_player2_mouse_index = "1"
input_player2_gun_trigger_mbtn = "1"
input_player2_gun_offscreen_shot_mbtn = "2"
input_player2_gun_start_btn = "6"
input_player2_gun_select_btn = "7"
input_player2_gun_dpad_up = "w"
input_player2_gun_dpad_down = "s"
input_player2_gun_dpad_left = "a"
input_player2_gun_dpad_right = "d"
input_player2_gun_aux_a = "f"
input_player2_gun_aux_b = "g"
input_player2_gun_aux_c = "h"

```

This maps most lightgun inputs (except for the left rear button) to the standard gun inputs of RetroArch to the buttons we've configured our lightguns to send.

This is good news! Now all emulators will receive the "correct" inputs!

How to map individual games

These steps might vary slightly from emulator to emulator, and sometimes even for games.

I found they work mostly verbatim for `lr-flycast`, `lr-fbneo` and `lr-duckstation`, and other emulators follow a similar mode. `lr-mame2003-plus` works with a bit of tinkering. It also works mainly in `lr-pcsx-rearmed`, but the gun's buttons are hard-coded to the mouse button inputs. This is fine for most games, but e.g. for `Resident Evil: Survivor`, the mapping did not suit me at all. So I used `lr-duckstation` instead, which worked fine right out of the gate, but does not support two lightguns at the same time.

Some tinkering in the `Options` menu of RetroArch might be required here and there, e.g. for gun offsets,

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)

[OK](#)

Border overlay and override files

Start the game you like to configure, get into, press **Hotkey + X** to get into the RetroArch menu. Go to **On screen overlay**, make sure the overlay is enabled, and select **SindenBorderWhiteThin.cfg** (you might have to exit a couple of folder levels if you have The Bezel Project or other bezels installed).

Note: If you feel that the gun is acting jittery (can happen if you're in a bright room or standing quite close to the screen), then you can also opt to use **SindenBorderWhiteMedium.cfg** instead, which is a bit broader. Exit the overlay menu.

Note: This is also the place where you can setup the 4:3 screen ratio or any shaders you might want to use for this game. I recommend choosing the shaders on a case-by-case basis, as e.g. barrel distortion can lead to the aim being off, and resource intense shaders might impact the game's performance.

Go to **Overrides** and select **Save core overrides** (if you have not configured any game for this core yet, just to create a file) and **Save game overrides**.

Exit the game (usually **Hotkey + Start**). If you've already set up the controllers as lightguns (see below), you absolutely need to restart the game, because for some reason retroarch loses the gun mappings after saving the remap/override.

Note: You could also create a config file with e.g. the overlay and your shaders, and **#include** it for each lightgun game's override instead. I just set it individually when starting the game for the first time, while testing everything out in it.

Lightgun mappings

If this is the first game for this core, go to the newly created core override config file, e.g.

/opt/retroPie/configs/all/retroarch/config/Flycast/Flycast.cfg for Flycast. Delete all of its contents (yes, you read that right) and replace them with just:

```
#include "/opt/retroPie/configs/all/retroarch/config/lightgun.cfg"
```

For Mame2003-Plus I use this, as some games map the buttons instead of the trigger/pump action (it does not conflict with controller mappings):

```
#include "/opt/retroPie/configs/all/retroarch/config/lightgun.cfg"
```

```
input_player1_a = "1"
input_player1_b_mbtn = "1"
input_player2_a = "f"
input_player2_b_mbtn = "1"
```

Note: Some games require special mappings/configs. For those, take a look at the BB8 compatibility list (<http://barebones.lightgun.dev/game-compatibility-list>). Duck Hunt for NES is one of them (so actually not ideal to start with). It requires the gun to be in Port 2. If you have two guns, no problem, just use the second one. If you only have one, you can add this to the game override you created earlier for the border (depending on your rom and core, this will have a path similar to **/opt/retroPie/configs/all/retroarch/config/FCEUmm/Duck Hunt (World).cfg** or to map it in all NES emulators you can use **/home/pi/RetroPie/roms/nas/Duck Hunt (World).zip.cfg**), which will map your gun to Port 2 instead of Port 1 (and vice versa, for that matter) for that particular game:

```
# Switch mouse
input_player1_mouse_index = "1"
input_player2_mouse_index = "0"
```

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

More information

OK

So much for Duck Hunt.

What we've set up will now load our lightgun input settings each time the core is loaded. This is needed, as the main retroarch config is overwritten by EmulationStation on each button remapping in the EmulationStation controller setup, and not all cores require our gun mappings. This way you can still use that way of configuring your controller in EmulationStation, but also add your lightgun mapping.

Now, start the game again. Go into the RetroArch menu as before, go up one level, and go to **Settings** -> **Input** and open Port 1. Scroll down and make sure the gun mappings are loaded. Do the same for Port 2. Now, go back to **Quick Menu** -> **Controls**. Go into Port 1 and select **lightgun** (or similar, e.g. **guncon** for PSX, **Zapper** for NES) - unless it is e.g. Duck Hunt on NES (see the note above). Do the same for Port 2 (if you have two lightguns or if you have mapped your gun to Port 2). Then select **Save Game Remap File**. Exit the game (after this configuration, the lightgun input mappings tend to get lost temporarily), and start it again.

Now you should be able to play it! Congrats!

Some convenience and pretty stuff

Pretty entries for Lightgun utilities in EmulationStation

To make the utility entries in the EmulationStation frontend a bit prettier, I downloaded this image from the Sinden homepage (<https://www.sindenlightgun.com/wp-content/uploads/2021/04/LNL010421-179-768x512.jpg>), named it **SindenImage.jpg** and copied it to `/home/pi/.emulationstation/downloaded_images/ports/`. Then edit `/opt/retroPie/configs/all/emulationstation/gamelists/ports/gamelist.xml` and add the following snippet at the bottom:

```
<game>
  <name>Sinden Lightgun Test P1</name>
  <image>~/emulationstation/downloaded_images/ports/SindenImage.jpg</image>
  <desc>Sinden Lightgun Test for Player 1</desc>
  <path>./SindenLightgunTestP1.sh</path>
  <name>SindenLightgunTestP1</name>
</game>
<game>
  <name>Sinden Lightgun Test P2</name>
  <image>~/emulationstation/downloaded_images/ports/SindenImage.jpg</image>
  <desc>Sinden Lightgun Test for Player 2</desc>
  <path>./SindenLightgunTestP2.sh</path>
  <name>SindenLightgunTestP2</name>
</game>
<game>
  <path>./SindenLightgunStartP1.sh</path>
  <name>Sinden Lightgun Start P1</name>
  <desc>Sinden Lightgun Start for Player 1</desc>
  <image>~/emulationstation/downloaded_images/ports/SindenImage.jpg</image>
</game>
<game>
  <path>./SindenLightgunStartP2.sh</path>
  <name>Sinden Lightgun Start P2</name>
  <desc>Sinden Lightgun Start for Player 2</desc>
  <image>~/emulationstation/downloaded_images/ports/SindenImage.jpg</image>
</game>
<game>
  <path>./StopSindenLightgun(s).sh</path>
  <name>Sinden Lightgun Stop</name>
  <desc>Sinden Lightgun Stop for both Players</desc>
  <image>~/emulationstation/downloaded_images/ports/SindenImage.jpg</image>
</game>
```

Run Lightgun drivers automatically

If you want your lightguns to start when you start a game, and stop them afterwards, you can use this method. Create a file `/opt/retroPie/configs/all/runcommand-onlaunch.sh` and add this content:

```
#!/bin/bash
/home/pi/RetroPie/roms/ports/SindenLightgunStartP1.sh
/home/pi/RetroPie/roms/ports/SindenLightgunStartP2.sh
```

Set `chmod +x` on the file to make it executable (i.e. run `chmod +x /opt/retroPie/configs/all/runcommand-onlaunch.sh`).

Similarly, create a file `/opt/retroPie/configs/all/runcommand-onend.sh`, and add this content:

```
#!/bin/bash
"/home/pi/RetroPie/roms/ports/StopSindenLightgun(s).sh"
```

Also set `chmod +x` on the file to make it executable.

Now the lightgun software is started before your emulators are launched, and stopped afterwards.

Patched lr-mame2016

Newer mames seem to have issues with lightgun support. `lr-mame2003-plus` works fine, but it does not support as many games as, e.g. `lr-mame2015`. There is a patched version by *MrLightgun*.

Edit `/home/pi/RetroPie-Setup/scriptmodules/libretrocores/lr-mame2016.sh` and find this snippet

```
function sources_lr-mame2016() {
    gitPullOrClone
}
```

Change it to read like this:

```
function sources_lr-mame2016() {
    # gitPullOrClone
    gitPullOrClone "$md_build" https://github.com/MrLightgun/mame2016-libretro.git
}
```

Now, as before with RetroArch, start `retroPie_setup` (run `sudo /home/pi/RetroPie-Setup/retroPie_setup.sh`), go to **Manage packages** -> **exp** -> **lr-mame2016**. Remove it if required, then **Install from source**. This will take quite a while. Afterwards, lightgun support for `lr-mame2016` should be patched.

More info on how to use this can be found in `SindenLightgunDocumentationV1.05_Linux.pdf` which comes with the latest stable download from the Sinden page (see above).

I personally have it patched, but I got all arcade games I wanted working with either `lr-fbneo`, `lr-fbalpha2012` or `lr-mame2003-plus`, so I'm not using it currently.

In closing

That was quite a ride! Hopefully, you could follow the instructions here without too much hassle. I'm not sure they work for every use-case. But they worked for mine. Thanks a lot to the nice people helping in the Sinden

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

[More information](#)

[OK](#)

Happy shooting!

DC/Naomi/Atomiswave

For Dreamcast / Naomi / Atomiswave setup for your Pi4 check out this page:

<https://sindenlightgun.miraheze.org/wiki/Flycast>

Retrieved from "https://sindenlightgun.miraheze.org/w/index.php?title=Raspberry_Pi_Setup_Guide&oldid=1191"

Content is available under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) unless otherwise noted.