

UPDATE: Based on **PlayBox** v2 Setup. Sinden LightGun v1.5b

Sinden Lightgun Documentation – Linux – Primarily Raspberry Pi 4 in this release

~~If you are using Pi4, it is recommended to use HarryDog's pre-built Pi4 image (minus roms) which is linked to on the Sinden Lightgun Discord.~~

This is a very rough guide that will hopefully get you started. As always the beta testers will have a go and help work out any gaps to make it easier. This build is focused on Pi4 specifically, Pi3B+ is also compatible with this release and other ARM devices such as Odroid XU4. Linux X86 is also now supported and it has its own folder.

The Linux software is a lot less user friendly than the Windows software, so to start with I recommend that you get to grips with the Sinden Lightgun in Windows and get an understanding of what border and settings works well on your display and what doesn't. Then use the same settings where possible on the Pi. The software will evolve to have more options and be more user friendly, a calibration screen is likely to be sorted soon.

This guide is centred around RetroPie but you can adapt to other environments if you have the knowledge, more support will follow.

~~Install RetroPie from the proper release and setup all your favourite games and emulators etc. You ideally need a controller connected to navigate all the UI. There are many many good guides on this.~~

~~Download the Sinden Lightgun Linux archive to your standard home folder on the Pi, (home/pi/) using FTP software.~~

~~You therefore have a folder called "SindenLightgun".~~

~~It has a folder for Player1 and a folder for Player2. If you use 2 lightguns you need to run an instance of each.~~

~~Firstly install the pre-requisites by running:~~

~~`./setup-lightgun.sh`~~

~~It is likely this will error saying it doesn't have permission so try:~~

~~`chmod +x setup-lightgun.sh`~~

~~Then run the above again, if you have issues try it with a sudo in front like this:~~

~~`sudo ./setup-lightgun.sh`~~

~~Hopefully everything will install OK. If you are a RetroPie user then you also need to run the following with similar instructions as above:~~

~~`sudo ./setup-retropie.sh`~~

~~Assuming you don't have lots of other kit setup then the configuration should already be setup. In the config change [SerialPortWrite](#) if you get a clash by probably increasing the number.~~

~~The RetroPie install script also copies some scripts into RetroPie/roms/Ports folder. This way you can start the Sinden Lightgun from within RetroPie. If your ports tab doesn't appear, try installing something like Wolfenstein3D or Prince Of Persia from RetroPie setup that will create a Ports tab for your scripts. There is obviously a way to do this without installing a game.~~

~~There is 2 ways to start a Lightgun instance:~~

mono LightgunMono.exe

Or

mono-service LightgunMono.exe

The difference is that mono-service runs in the background. Currently there is not a graceful way to stop Sinden Lightguns other than the kill script ~~inside Ports~~. This works but then you can't start them again without a reboot.

Update, the ability to restart should now be fixed in v1.5 of the software.

In the ~~Ports tab~~ there is also "SindenLightgunTest This is just a very simple test harness for you to see that the Sinden Lightgun is working.

[PLEASE NOTE this actually runs the lightgun too so you need to not start the lightgun before running this test.](#)

This screen will eventually become a calibration screen.

Drag the cursor to the bottom right to exit it ends after a couple of minutes.

Occasionally with a current bug it might just keep streaming text to the console till the 2 minutes ends. You should see the cursor responds very fast, this is because there is not the overhead of windows or a game controlling the cursor. It gets the result and draws it as quick as it can.

If you are typing things into the console I recommend a remote ssh session using Putty ~~which just needs the ip address.~~

So now lets try to play some games in RetroPie.

~~Firstly there is a very annoying bug in RetroArch currently on the Pi4, it utilises the X11 code branch for lightgun support and does a weird thing where the cursor is slightly off to the side BUT as soon as you go all the way to the left hand screen it lines up. I haven't been able to work out why, but the solution is to use the non X11 code branch for the lightgun support. So you can leave it as it is and just remember to shuffle off to the left side everytime you start a game or you can use my patched version of RetroArch which is just switching a single file, but please remember to back up your original one. The patched file is in the archive with instructions.~~

~~***Update*** I'm struggling to recreate the bug described above and "think" it may have been fixed in the latest RetroArch, but I'm including the patch regardless for reference. Try updating to the latest version of RetroArch if you get the issue.~~

To start with, this guide is aimed at people who have used a Wii Remote or Aimtrak or similar in RetroPie, it will evolve more friendly, but now you can just use the lightguns like you do any mouse device.

Assign the trigger and buttons to the RetroArch Lightgun settings and get the mouse index right. On my system it is Player1 mouse index = 0 and Player 2 = 3.

Add the template Sinden Lightgun borders from the archive to the RetroArch overlay folder:
~~/opt/retropie/emulators/retroarch/overlays~~

Now in your relevant emulator you should be able to play. Start with Duck Hunt on FCEUMM which is for NES. Go into the menus using your hotkeys.

Go into options make sure the Zapper type is set

to "lightgun" and for now "show crosshair".

~~Then go back a level and go to On-Screen overlay. Add a Sinden Lightgun border under Overlay preset. I recommend SindenBorderWhiteMedium to start with. Opacity should be on 1. You will want to save your settings under overrides, save them for the core.~~

In most emulators you also need to set the controller mode to a lightgun but usually FCEUMM doesn't need it.

Remember in some systems it is the second controller that needs to be set like the **Superscope** on snes.

~~If your game screen doesn't match the border you need to go to retroarch settings, video and scaling and set the scaling to 4:3, then it should match perfectly.~~

This are the emulators that work out of the box:

FCEUMM/NESTOPIA – NES

SNES9X – SNES

GENESIS-PLUS-GX – Master System / MegaDrive/Genesis and SegaCD

Lr-PCSX-Rearmed – Playstation 1 Guncon games or patched Justifier games

Lr-flycast – Dreamcast and Sega Naomi including HOTD2

Lr-Mame2016 – Arcade games

Lr-opera – 3DO probably best way to play American Laser Games, but has a reload bug that prevents progress on Mad Dog McCree. Should be fixable soon.

So there are a few flaws which I have fixed in the emulators but you need to install from my github.

Genesis-plus-gx: The inputs like trigger don't work on Genesis/Megadrive but do on Master System.

This technique applies to all of the adapted emulators. Go to /RetroPie-Setup/scriptmodules then find the relevant emulator install script under libretrocores or emulators.

Edit the .sh file and change the install source from github...libretro....game to github...libretro....game.

Like this:

```
function sources_lr-genesis-plus-gx() {  
    gitPullOrClone "$md_build" https://github.com/sindenlightgun/Genesis-Plus-G  
}
```

Then on RetroPie, go to the RetroPie tab, do RetroPieSetup -> Manage packages -> Manage main packages -> lr-genesis-plus-gx then select remove, then install from source. This one takes about 3 minutes I think.

Now hopefully the game works.

Currently offscreen reload is broken in either Dreamcast or Naomi and I can't remember which but it's fixed on my branch.

And offscreen shoot for selecting menus is fixed in my version of FCEUMM but you can just use select on your controller instead.

So the remaining thing is Mame. AdvMame has good lightgun support but doesn't work properly for me on Pi4. There is some kind of memory leak which bezels/overlays that is causing it to crash. However I couldn't get Lightguns to work properly in the other versions. So for now I have hacked support into Mame2016. Install that using the method outlined above. Obviously will look to improve this situation. The github is **"mrlightgun"** not "sindenlightgun" this time.

~~Use RetroArch to add the border in the usual way.~~

If you use my version it maps the lightgun to JoypadRXaxis and JoypadRYAxis basically the right control stick. So you need to assign controls to that. However it doesn't always get picked up when you go into tab in Mame. So just assign you x and y to something else other than default. Then edit the config located here (using Terminator2 as example):

/RetroPie/roms/arcade/mame2016/cfg/term2la.cfg

Here I have assigned the analog to the Joystick Right Axis which is hacked to be our lightguns:

```
<port tag=":STICK0_X" type="P1_AD_STICK_X" mask="255" defvalue="128$
    <newseq type="standard">
        JOYCODE_1_RXAXIS
    </newseq>
</port>
<port tag=":STICK0_Y" type="P1_AD_STICK_Y" mask="255" defvalue="128$
    <newseq type="standard">
        JOYCODE_1_RYAXIS
    </newseq>
</port>
<port tag=":STICK1_X" type="P2_AD_STICK_X" mask="255" defvalue="128$
    <newseq type="standard">
        JOYCODE_2_RXAXIS
    </newseq>
</port>
<port tag=":STICK1_Y" type="P2_AD_STICK_Y" mask="255" defvalue="128$
    <newseq type="standard">
        JOYCODE_2_RYAXIS
    </newseq>
```

Player2 does work also as Joypad 2 RXaxis and RYaxis if you have correctly set the mouse index in retroarch but not picking up the mouse buttons for some reason, not worked out yet. You can assign another action to the trigger like a keyboard press which you can use in the short term if very keen but that will mess up other games. I would maybe use a button to shoot instead for 2nd player on Mame rather than re-assign the trigger.

If you need to change Sinden Lightgun settings you need to edit the following file like so:

nano LightgunMono.exe.config

Should be obvious what a lot of them do, copy your values from windows. You might want to change the resolution to 320x240 if you want to use less processing or want a fractionally faster response. I think 640x480 performance is the correct setting as a default.

To calibrate

Inside Ports, select "SindenLightgunTestP1" or P2, then follow the instructions on the screen. Hold down DPad Left for 3 seconds till the cursor goes central, then shoot the cursor to calibrate or pump action to cancel.

To change the key mapping in the config, here is the mapping of what means what number, some of these are windows specific, such as border, the defaults are the same as in windows, offscreen reload is turned off by default as most cores have their own offscreen reload built in:

```
Buttons = new List<ButtonAssignment>();
    //ButtonAssignment ba = new ButtonAssignment("None", 0);
    Buttons.Add(new ButtonAssignment("None", 0));
    Buttons.Add(new ButtonAssignment("MouseLeft", 255));
    Buttons.Add(new ButtonAssignment("MouseMiddle", 254));
    Buttons.Add(new ButtonAssignment("MouseRight", 253));
    Buttons.Add(new ButtonAssignment("Pause Movement", 252));
    Buttons.Add(new ButtonAssignment("Turbo Fire MouseLeft", 251));
    Buttons.Add(new ButtonAssignment("Turbo Fire/Reload MouseLeft/Right",
250));
    Buttons.Add(new ButtonAssignment("Border On/Off (Alt B)", 249));

    Buttons.Add(new ButtonAssignment("0", 48));
    Buttons.Add(new ButtonAssignment("1", 49));
    Buttons.Add(new ButtonAssignment("2", 50));
    Buttons.Add(new ButtonAssignment("3", 51));
    Buttons.Add(new ButtonAssignment("4", 52));
    Buttons.Add(new ButtonAssignment("5", 53));
    Buttons.Add(new ButtonAssignment("6", 54));
    Buttons.Add(new ButtonAssignment("7", 55));
    Buttons.Add(new ButtonAssignment("8", 56));
    Buttons.Add(new ButtonAssignment("9", 57));

    Buttons.Add(new ButtonAssignment("A", 65));
    Buttons.Add(new ButtonAssignment("B", 66));
    Buttons.Add(new ButtonAssignment("C", 67));
    Buttons.Add(new ButtonAssignment("D", 68));
    Buttons.Add(new ButtonAssignment("E", 69));
    Buttons.Add(new ButtonAssignment("F", 70));
```

```

Buttons.Add(new ButtonAssignment("G", 71));
Buttons.Add(new ButtonAssignment("H", 72));
Buttons.Add(new ButtonAssignment("I", 73));
Buttons.Add(new ButtonAssignment("J", 74));
Buttons.Add(new ButtonAssignment("K", 75));
Buttons.Add(new ButtonAssignment("L", 76));
Buttons.Add(new ButtonAssignment("M", 77));
Buttons.Add(new ButtonAssignment("N", 78));
Buttons.Add(new ButtonAssignment("O", 79));
Buttons.Add(new ButtonAssignment("P", 80));
Buttons.Add(new ButtonAssignment("Q", 81));
Buttons.Add(new ButtonAssignment("R", 82));
Buttons.Add(new ButtonAssignment("S", 83));
Buttons.Add(new ButtonAssignment("T", 84));
Buttons.Add(new ButtonAssignment("U", 85));
Buttons.Add(new ButtonAssignment("V", 86));
Buttons.Add(new ButtonAssignment("W", 87));
Buttons.Add(new ButtonAssignment("X", 88));
Buttons.Add(new ButtonAssignment("Y", 89));
Buttons.Add(new ButtonAssignment("Z", 90));

Buttons.Add(new ButtonAssignment("a", 97));
Buttons.Add(new ButtonAssignment("b", 98));
Buttons.Add(new ButtonAssignment("c", 99));
Buttons.Add(new ButtonAssignment("d", 100));
Buttons.Add(new ButtonAssignment("e", 101));
Buttons.Add(new ButtonAssignment("f", 102));
Buttons.Add(new ButtonAssignment("g", 103));
Buttons.Add(new ButtonAssignment("h", 104));
Buttons.Add(new ButtonAssignment("i", 105));
Buttons.Add(new ButtonAssignment("j", 106));
Buttons.Add(new ButtonAssignment("k", 107));
Buttons.Add(new ButtonAssignment("l", 108));
Buttons.Add(new ButtonAssignment("m", 109));
Buttons.Add(new ButtonAssignment("n", 110));
Buttons.Add(new ButtonAssignment("o", 111));
Buttons.Add(new ButtonAssignment("p", 112));
Buttons.Add(new ButtonAssignment("q", 113));
Buttons.Add(new ButtonAssignment("r", 114));
Buttons.Add(new ButtonAssignment("s", 115));
Buttons.Add(new ButtonAssignment("t", 116));
Buttons.Add(new ButtonAssignment("u", 117));
Buttons.Add(new ButtonAssignment("v", 118));
Buttons.Add(new ButtonAssignment("w", 119));
Buttons.Add(new ButtonAssignment("x", 120));
Buttons.Add(new ButtonAssignment("y", 121));
Buttons.Add(new ButtonAssignment("z", 122));

Buttons.Add(new ButtonAssignment("Return", 176));
Buttons.Add(new ButtonAssignment("Space", 32));
Buttons.Add(new ButtonAssignment("Escape", 177));
Buttons.Add(new ButtonAssignment("Tab", 179));
Buttons.Add(new ButtonAssignment("Up", 218));
Buttons.Add(new ButtonAssignment("Down", 217));
Buttons.Add(new ButtonAssignment("Left", 216));
Buttons.Add(new ButtonAssignment("Right", 215));

Buttons.Add(new ButtonAssignment("+", 43));
Buttons.Add(new ButtonAssignment(",", 44));
Buttons.Add(new ButtonAssignment("-", 45));
Buttons.Add(new ButtonAssignment(".", 46));

```

```
Buttons.Add(new ButtonAssignment("F1", 194));
Buttons.Add(new ButtonAssignment("F2", 195));
Buttons.Add(new ButtonAssignment("F3", 196));
Buttons.Add(new ButtonAssignment("F4", 197));
Buttons.Add(new ButtonAssignment("F5", 198));
Buttons.Add(new ButtonAssignment("F6", 199));
Buttons.Add(new ButtonAssignment("F7", 200));
Buttons.Add(new ButtonAssignment("F8", 201));
Buttons.Add(new ButtonAssignment("F9", 202));
Buttons.Add(new ButtonAssignment("F10", 203));
Buttons.Add(new ButtonAssignment("F11", 204));
Buttons.Add(new ButtonAssignment("F12", 205));
```

Mapping buttons to RetroArch

You can access RetroArch controls under the RetroPie tab -> RetroArch settings.

You need to assign the relevant lightgun Player 1 controls to the lightgun. I recommend the following (for a right hander):

Trigger -> Left Mouse

AuxA -> Right Mouse

AuxB -> Left Mouse

Gun Start -> Dpad Up

Gun Select (usually credit) -> Dpad Left

Gun Down -> Dpad Down

Gun Right -> Dpad Right

Offscreen Reload -> Front Right

AuxC -> Back Right

If you use SNES you might want to assign turbo but you can do that on your control pad instead. Also in SNES as standard Trigger and AuxA are usually reversed so you might want to swap them inside the Snes9X core. I believe it is due to the way the SuperScope was designed as a bazooka so trigger was on top.

Recoil

The Recoil functionality mirrors the windows settings:

<add key="IAgreeRecoilTermsInLicense" value="0"/> Read the terms in License.txt and set to 1

<add key="EnableRecoil" value="0"/> Set to 1 to enable

<add key="RecoilTrigger" value="1"/> Enable recoil on trigger

<add key="RecoilTriggerOffscreen" value="0"/> Enable recoil on trigger offscreen

<add key="RecoilPumpActionOnEvent" value="0"/>

<add key="RecoilPumpActionOffEvent" value="0"/>

<add key="RecoilFrontLeft" value="0"/>

<add key="RecoilFrontRight" value="0"/>

<add key="RecoilBackLeft" value="0"/>

<add key="RecoilBackRight" value="0"/>

<add key="RecoilStrength" value="100"/>

<add key="TriggerRecoilNormalOrRepeat" value="1"/> Set to 0 for single shot, 1 for automatic

<add key="AutoRecoilStrength" value="80"/> Strength of recoil pulses 0-100

<add key="AutoRecoilStartDelay" value="5"/> Delay after first shot so it doesn't repeat fire when firing a single shot

<add key="AutoRecoilDelayBetweenPulses" value="13"/> How much time between pulses, so therefore speed.

Fedora

If you are using Fedora, the included script to build libjpeg is not compatible, this alternative seems to work (credit to TLP):

```
sudo dnf copr enable lyra/libjpeg
```

```
sudo dnf install libjpeg8
```

so like this:

```
sudo dnf install -y mono-complete
```

```
sudo dnf install -y v4l-utils
```

```
sudo dnf install -y SDL-devel
```

```
sudo dnf install -y SDL_image-devel
```

```
sudo dnf copr enable -y lyra/libjpeg
```

```
sudo dnf install -y libjpeg8
```

Hopefully this is enough to get you going but it is very raw.