

Instructions For Connecting To PSX or PS2 as Guncon45

Please note, the aim is to make this process super easy in the future but to start with for now it is not recommended for non-technical people.

Please note I am not responsible for anything that goes wrong with this mod, this is a guide for people that fully understand this kind of electronics.

This guide will improve when more people have had a chance to experiment and input.

You will need:

Raspberry Pi (4 is recommended, but 3B+ should work too)

Arduino Uno R3 with a USB cable

Dupont Wires Male to Male

Sinden Lightgun

OSSC (alternatively a Micro MinimOSD when written up)

PS2 or PS1 (recommend PS2)

A way to connect Playstation through OSSC, recommend component

You need to download at least version 1.6 of the Sinden Lightgun software and upgrade your firmware using the windows software to version 1.6.

This guide assumes you have some knowledge of installing the Sinden Lightgun software on the Pi, if you don't please see the Linux instructions PDF in the software archive. I recommend using a stock RetroPie build or the Sinden Lightgun basic RetroPie build put together by HarryDog on the Sinden Lightgun Discord group.

Install Arduino IDE, you need this to install the firmware on the Arduino Uno R3. Open SindenLightgunPSXDemoPalV1.ino (or NTSC) in the IDE and connect the Arduino Uno R3 via USB to your PC and upload the sketch.

You need to connect the Arduino correctly to the PS1 or PS2, please see Appendix A for some ideas on how to do this.

Get your Playstation displaying correctly on your TV via the OSSC. To show a white border you need to use a feature called "mask":

Menu

Post-Proc

Horizontal Mask -> I use 29 pixels

Vertical Mask -> I use 13 pixels

I also tell my TV to display in 4:3 as the PS2 output I get from the OSSC is stretched.

Connect the Raspberry Pi to the mains and boot, set it up like you would if playing emulators (see Linux instructions). For ease of use set it up on the network so you can use a remote command line.

Connect your Sinden Lightgun, this is to make sure it gets the lowest serial identity /dev/ttyACM0. Then connect the Arduino which should get /dev/ttyACM1. If automating from boot you need to make sure you know which is which. You can use udev rules to force the assignment also.

Using the command line go to the player1 folder. Edit the config with "nano LightgunMono.exe.config".

Set the SerialPortWrite to be /dev/ttyACM0 and set SerialPortSecondary to "/dev/ttyACM1". NOTE this will mess up this working with your emulators so you might want to install a second folder of Sinden Lightgun software for this. On boot I found that the Arduino always liked to grab ACM0 so you might want to swap the numbers round.

In summary when you set the SerialPortSecondary it exports serial data every frame with information from the lightgun:

Byte1 = 222 (static header value)

Byte2 = 0-256 X Coordinates

Byte3 = 0-256 Y Coordinates

Byte4 = A byte with individual bits describing if

Trigger/PumpAction/FrontLeft/FrontRight/BackLeft/BackRight are on or off

Byte5 = A byte with individual bits describing if

DPad Up/Down/Left/Right arrive on or off

Byte6 = 223 (static footer value)

If you set SerialPortSecondary to the Raspberry Pi GPIO serial port then it will output this serial information on the GPIO serial pins. This might be a useful way to connect to something like a MiSTer. I will do another simple ProMicro sketch soon that will output USB mouse, which will enable more compatibility as you can connect using the serial pins and output the mouse on the ProMicro USB.

The PSX sketch reads these serial values that are coming through the USB and it turns them into SPI signals that go into the Playstation.

If you start the Sinden Lightgun software "mono LightgunMono.exe" then everything should hopefully work. Don't worry if you get occasional JPEG error messages, due to the speed of transfer of so much video you sometimes get bad frames on the Raspberry Pi.

You can attach a pedal to the Arduino on Pins 3 or 4 which correspond to the side buttons on the Guncon45. Pin 5 you can attach a trigger but I'm not sure how helpful that is except for testing.

I recommend Time Crisis 2 to get things setup well. On the calibration screen of TC2 you can check your alignment. Usually the Y axis is almost spot on but the X axis will probably have a little drift. You can adjust this in the Sinden Lightgun config. "nano LightgunMono.exe.config".

I use

OffsetXRatio = "1.07"

This lines it up perfectly for me although remember I am using PAL.

Also don't forget to calibrate your lightgun in the Sinden Lightgun software to get your Sinden Lightgun calibrated to start with. Try to do this using a game on RetroPie like the cursor in DuckHunt and don't change it when looking at a cursor in Time Crisis or you might get strange results.

I haven't tested NTSC yet so we may need to tweak some of the Arduino sketch.

If you are successfully running all the software, when you boot Time Crisis 2 it should go to the

calibration screen. If it doesn't, it means something is not connected correctly or the software is not running correctly.

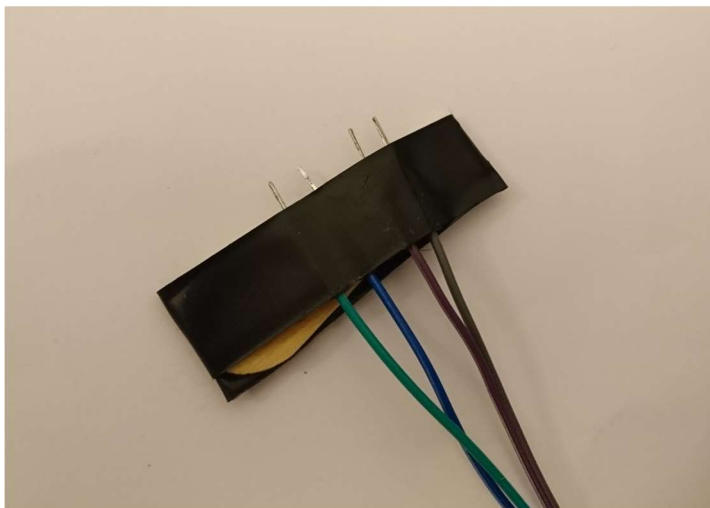
I don't recommend trying to change the pins on the Arduino of what does what as you are likely to come unstuck.

How to connect to the PSX controller port

This is the hardest part of the mod but hopefully you can get a basic version setup easily. Here are 3 suggestions on how to do it:

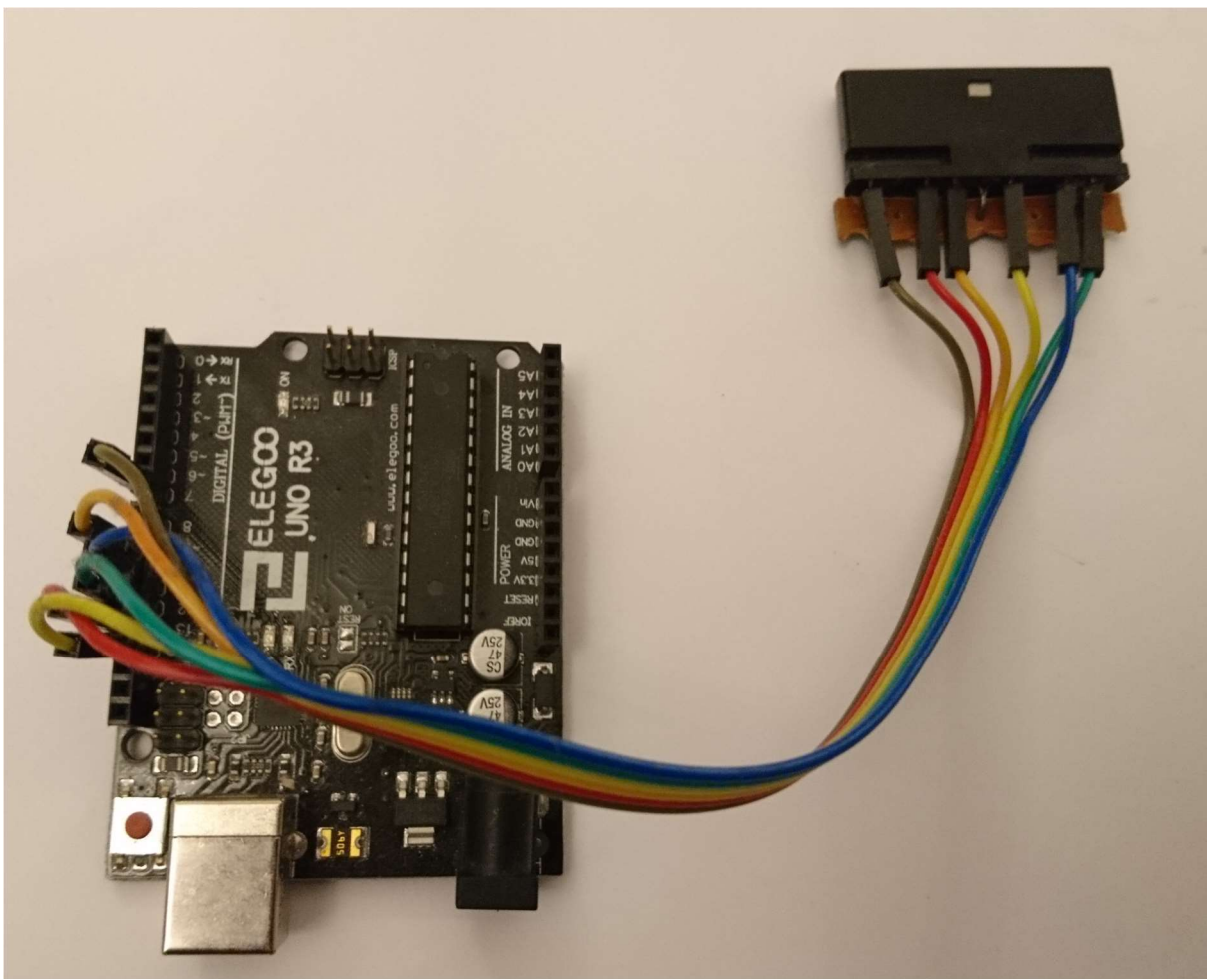
A) The ice lolly stick hack – a shameful dirty hack....

Get an ice lolly stick or a similar flat piece of wood. Cut it to be about the same length as a PSX controller port. Lay the Dupont wires in place on top of it, in line with the correct holes. Stick insulating tape across the top to hold in place. When you plug it in, the weight of the unsupported lolly stick tends to pull downwards which makes the pins go up and make a good contact. This picture is just shown for context, but it needs to be something similar to this that lines up with 6 wires.



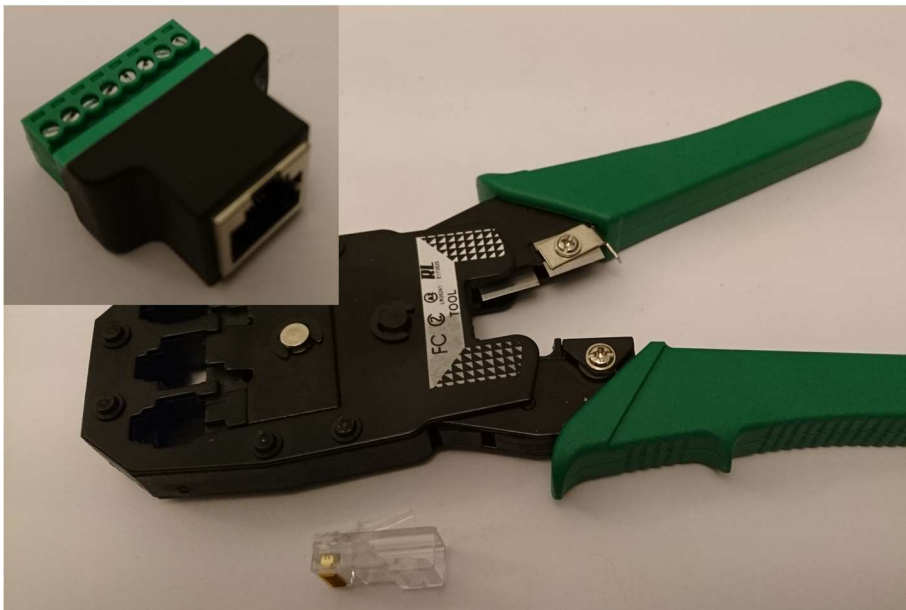
B) Wireless dongle

By a budget low cost wireless PS2 controller with dongle. Smash open the dongle and re-use it as a connector. With any luck you can solder a really good connector. With the one I have shown here it was possible to slot the Dupont pin into the connector and get it to hold.



C) PSX extension cable and Ethernet plug

Buy a PSX extension cable, cut off the end that doesn't go into the Playstation. Expose the wires, I think there are only normally 7 or 8 I think. Get an ethernet plug and a crimping tool, crimp the Ethernet plug on the end. Get an Ethernet breakout board, a little PCB that has an Ethernet socket and breaks the wires out to pins. You can then easily connect wires to the Arduino pins. This solution is best because the end that goes into the Playstation is properly moulded.



This is the wiring:

As you look at the Playstation port, there are 9 pins, left to right:

- 1 ACK – Arduino D7
- 2 Not Connected
- 3 CLOCK – Arduino D13/SCK
- 4 ATT – Arduino D10/SS
- 5 VCC – Not used
- 6 GND – Arduino Ground
- 7 N/C (9 Volts unused)
- 8 COMMAND – Arduino D11/MOSI
- 9 DATA – Arduino D12/MISO

So 6 wires connected:

1-3 4-6 -89