

AD

AD

W

F

SEA

AN

al Site

Sign Up

S

Setting Up PlayStation Controllers on the Raspberry Pi

Feb 17, 2022

 Gaming Guides

If

If you're looking to connect a PlayStation controller with the Raspberry Pi, you may have found it's not very simple to do so. Fortunately, this tutorial will help simplify the process.

AD



Our Raspberry Pi PlayStation controller guide will walk you through the process of setting up and connecting both PlayStation 3 controllers and PlayStation 4 controllers.

For PS3 controllers, we show you how to install the six-pair software and utilize it to set up your PS3 controller so it can connect to whatever Bluetooth device you are using on your Raspberry Pi.

In this PS4 controller tutorial, we walk you through the process of pairing it with the [Raspberry Pi's Bluetooth](#) and also offer alternative solutions if your controller doesn't work with the standard Bluetooth stack.

You can also set up [Xbox controllers on the Raspberry Pi](#), as well as several other types of game controllers.

AD

NFL Sunday Ticket

Ad By YouTube & YouTube TV

Sign up now

Setting up controllers is the perfect way to have your [Pi ready to play some games](#). You might also be interested in using the controllers in other Raspberry Pi projects that make use of a gamepad.

search text



SHOW ALL

AD

Equipment List

AD

Below are all the pieces of equipment that I made use of for this setting up PlayStation controllers on the Pi tutorial.

Recommended

-  [Raspberry Pi](#) 2 or newer
-  [Micro SD Card](#)
-  [Power Supply](#)
-  [PlayStation Controller](#) (PS3 Controller or PS4 Controller)
-  [Bluetooth Adapter](#) (Only if not using a Pi 3 or Zero W)

Optional

-  [Raspberry Pi Case](#)

AD



The advertisement features the JC Penney logo in a red box on the left. In the center, the text "Let's Get Ready to Move" is displayed in large, bold, yellow letters, with the tagline "Comfy every day in every way from class to courts" underneath in smaller white text. On the right, there is a "Shop Now" button.

AD

Setting up a PS3 Controller on the Raspberry Pi

Setting up a PlayStation 3 controller on the Raspberry Pi isn't an entirely straightforward process. Especially if you intend on using it wirelessly, we will walk you through all the steps to get it up and running in no time.

AD



**Ukrainian children need
your support.**

DONATE NOW

If you want to use your PS3 Controller wired, you should be able to use it just as a plug and play device.

However, we recommend you go to our section on compiling and installing sixad. Sixad is a piece of software that ensures that all of the PS3 Controllers features are supported.

Before you begin this section of the Raspberry Pi PlayStation tutorial make sure that you have a USB mini cable handy as you will require this to pair the controller with the Raspberry Pi. The reason for this is due to the way the PS3 Controller is set up to talk with Bluetooth controllers.

Preparing your PS3 Controller to Connect to the Raspberry Pi

AD



**Give a brighter future
to Ukrainian orphans**

FIND OUT HOW

AD

1. Before we get started with installing anything on our Raspberry Pi, we must first run the following two commands.

These commands will ensure that [your Raspberry Pi is up to date](#).

```
Terminal $
```

 Copy

```
sudo apt update  
sudo apt upgrade
```

2. To begin we first need to install a package, this is so we can compile the sixpair software. This package being the libusb-dev, this is for programs to interact with USB devices.

```
Terminal $
```

 Copy

```
sudo apt install libusb-dev
```

3. With our required package now installed we can go ahead and download the sixpair code and make a folder for it.

```
Terminal $
```

 Copy

```
mkdir ~/sixpair  
cd ~/sixpair  
wget http://www.pabr.org/sixlinux/sixpair.c
```

AD

NFL Sunday Ticket

Ad By YouTube & YouTube TV

Sign up now

- 4.** Now with the sixpair code downloaded to our Raspberry Pi, we need to compile it, to do this we can just run the following command.

This command will tell the GCC compiler to compile the code.

Terminal \$

 Copy

```
gcc -o sixpair sixpair.c -lusb
```

- 5.** With Sixpair now compiled on our Raspberry Pi, we need to plug our PS3 Controller into the Raspberry Pi using the USB mini cable.

- 6.** Once the controller has been plugged in, we can run sixpair by running the command below.

Sixpair will re-configure the controller so that it will talk with our Bluetooth device.

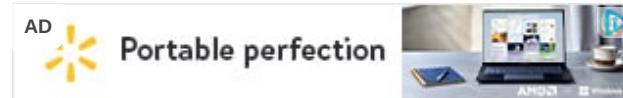
Terminal \$

 Copy

```
sudo ~/sixpair/sixpair
```

AD

7. If the sixpair software has successfully reconfigured the PS3 Controller to talk with your Raspberry Pi's Bluetooth dongle, then you should see something like below appear on your command line.



Output

```
Current Bluetooth master: b8:27:eb:a4:59:08  
Setting master bd_addr to b8:27:eb:a4:59:08
```

8. Now at this point, you are required to unplug the PS3 controller from the Raspberry Pi, now the changes have been made we should no longer need to plug your controller in by USB unless your Bluetooth adapter changes.

You now have two options, these being compiling and installing sixad to handle the connection between the PS3 controller or just using the bluetoothctl tool to make a connection to the PS3 Controller.

Personally, we recommend you try using Bluetooth as it is the most compatible with other Bluetooth devices.

Continue to the next section for how to pair your PS3 controller by using Bluetooth. Otherwise, skip to the section after for how to use SIXAD to pair the controller.

^{AD}Using bluetoothctl to connect your PS3 Controller

- With the PS3 Controller now ready to be connected with our Raspberry Pi we can move onto actually pairing it with the Raspberry Pi.



To do this, we need to first start up the Bluetooth configuration tool by running the following command on the Raspberry Pi.

Terminal \$

Copy

```
sudo bluetoothctl
```

- Now that we have launched up the Bluetooth configuration tool, we need to switch the agent on and set it to utilize the default-agent. Just type in the following two commands to do this.

Terminal \$

Copy

```
agent on  
default-agent
```

AD

5. Once we have enabled the agent we now want to actually scan for new Bluetooth devices, running the following command will tell the tool to start scanning.

```
Terminal $
```

 Copy

```
scan on
```

AD

You will notice immediately that any Bluetooth device in the area will start showing up in your command line.

Ignore these for now as we only want to pay attention to the PS3 controller itself.

4. Now is the time to press down the PlayStation button on the controller, so it begins to talk to the Raspberry Pi. You should soon start to see entries appear in the command line.

AD



5. Once you see something like the following appear, make sure you copy down the MAC address that is displayed. It is the sequence of text that is separated with ‘:’ symbols.

Output

```
[NEW] Device B8:27:EB:A4:59:08 B8-27-EB-A4-59-08
[CHG] Device B8:27:EB:A4:59:08 Connected: no
[DEL] Device B8:27:EB:A4:59:08 B8-27-EB-A4-59-08
```

AD



Children
of Heroes
Charity fund

Give a **brighter future**
to **Ukrainian orphans**

FIND OUT HOW

6. Now that you have a MAC address available copy it down, in our example, this is “**B8:27:EB:A4:59:08**”.

With the MAC address handy you want to use that in the next command. With this following command, we are trying to get a successful connection and a UUID returned by the device.

Please note you may have to run this command a few times before it is successful.

Make sure that you replace “**YOUR_MAC_ADDRESS**” with the one you found in **step 7**.

Terminal \$

 Copy

```
connect YOUR_MAC_ADDRESS
```

Please keep an eye on your PlayStation 3 controller, if it stops trying to connect then make sure you press the PS button again.

7. You will know a successful connection has been made when you see the text as we have shown below appears on the command line. With the **UUIDs** available we can proceed to the next step.

Output

```
Attempting to connect to B8:27:EB:A4:59:08
[CHG] Device B8:27:EB:A4:59:08 Modalias: usb:v054Cp0268d0100
[CHG] Device B8:27:EB:A4:59:08 UIDs:
00001124-0000-1000-8000-00805f9b34fb
00001200-0000-1000-8000-00805f9b34fb
```

AD



- 8.** Now that you have the MAC address we need to add this to our Bluetooth trusted list. What this does is allow the controller to connect to the Raspberry Pi on its own.

To do this, we need to just type in the following command to our Raspberry Pi. Make sure that you replace “**YOUR_MAC_ADDRESS**” with the one you found in **Step 7**.

```
Terminal $
```

Copy

```
trust YOUR_MAC_ADDRESS
```

- 9.** Once your PS3 controller has been successfully added to the trusted list, you should see something like below appear on the command line.

```
Output
```

```
[CHG] Device B8:27:EB:A4:59:08 Trusted: yes  
Changing B8:27:EB:A4:59:08 trust succeeded
```

- 10.** With your PS3 controller now successfully connected to your Raspberry Pi, we can now quit out of the Bluetooth configuration tool.

→ quit out of the tool you can either use the command below or just press **CTRL + D**.



Terminal \$

Copy

```
quit
```

11. Now you can restart the Raspberry Pi. We do this so we can test to make sure everything is working as intended. Run the following command to reboot the Raspberry Pi.

Terminal \$

Copy

```
sudo reboot
```

12. With the Raspberry Pi now restarted, press the PS button on your controller to test the connection between it and the Raspberry Pi.

The LED's on the controller should briefly flash as it connects, afterward however if everything has worked correctly only one LED should remain lit on the controller.

^{AD}Using SIXAD to Connect your PS3 Controller

1. Before we get started with compiling and setting up SIXAD on our Raspberry Pi, we must first install a few packages that we require.



The essential packages here are libusb-dev and libbluetooth-dev as they contain the code that we need to compile the sixad software.

Libusb-dev contains code that allows us to interact with USB devices that are plugged into the computer through code.

Libbluetooth-dev provides the code required to be able to interact with the operating systems Bluetooth stack. In sixad's case this is used to detect when a PS3 device is connected to the Raspberry Pi.

Terminal \$

Copy

```
sudo apt install git libbluetooth-dev checkinstall libusb-dev
sudo apt install joystick pkg-config
```

2. Now that we have all the packages that we could need let's go ahead and clone the SIXAD repository from Retropie, the reason we use Retropie's fork is that it has a few improvements and is tested on the Raspberry Pi.

Run the following two lines on your Raspberry Pi to clone the GIT repository.

AD Terminal \$

 Copy

```
cd ~  
git clone https://github.com/RetroPie/sixad.git
```

AD

- 3.** With the repository now cloned to our Raspberry Pi, we can now move onto actually compiling it. Compiling SIXAD is simple to do and just requires to run the following two commands.

The first command moves us into the folder, the second line runs the makefile and compiles the program.

Terminal \$

 Copy

```
cd ~/sixad  
make
```

- 4.** Now with the sixad software now compiled there are a couple more things to be done before we can use it.

The first of these is to make a folder where sixad keeps track of all its profiles.

Run the following command to create a folder for it.

AD Terminal \$

 Copy

```
sudo mkdir -p /var/lib/sixad/profiles
```

5. Finally lets run the `checkinstall` package that we installed in **step 1** of this section.

AD



**Ukrainian children need
your support.**

DONATE NOW

Basically what this tool does is check the current install then proceeds to run the commands “`make install`” if they are required.

Terminal \$

 Copy

```
sudo checkinstall
```

At specific points, you will be asked a few things, ignore these and just press `ENTER` to continue until the installation is complete.

Once the installation is complete, you should see the following appear in your command line.

AD Output

Done. The new package has been installed and saved to

/home/pi/sixad/sixad_20171230-1_armhf.deb

You can remove it from your system anytime using:

dpkg -r sixad

AD



**Ukrainian children need
your support.**

DONATE NOW

6. Now that we have finished installing sixad we can finally run it by typing the following command in on our Raspberry Pi.

Terminal \$

Copy

sudo sixad --start

If everything is working as intended, you will now be asked to press the PS button on your controller. Do this now to connect your controller to your Raspberry Pi.

AD



7. Now that we know that sixad is up and running we want to make sure that it starts at boot so we can always connect our PS 3 controller.



To do this, we can run the following command to add sixad to the boot sequence of the Raspberry Pi.

Terminal \$

Copy

```
sudo update-rc.d sixad defaults
```

AD

Setting up a PlayStation 4 Controller On the Raspberry Pi

There are a few different ways you can utilize your PlayStation 4 controller with your Raspberry Pi. Some of these are simpler to deal with than others.

Sony USB Wireless Adapter

The Sony USB Wireless Adapter is probably the easiest solution to getting your PS4 controller paired with the Raspberry Pi. Especially since the dongle should be able to work out of the box and pair with any official PS4 Controllers.

Bluetooth

AD

Bluetooth is a bit more complicated than just using the official Sony USB Wireless adapter and requires you to have a Raspberry Pi 3, Raspberry Pi Zero W, or a USB Bluetooth adapter.

However, this method does not work correctly for every PS4 controller, and you may have to end up utilizing a userspace controller driver called ds4drv to be able to interact with it.

Please note that if you previously set up the SIXAD tool, you will have to uninstall it to use your PS4 controller with the Raspberry Pi.

USB Cable

AD Using a USB Cable is the easiest way to utilize your PS4 Controller, just plug it in with a Micro USB cable like you would when first connecting your controller to the PS4 itself.

Using the USB connection, the PS4 controller should just work straight off the bat.

Pairing your PS4 Controller via BluetoothCTL

Please note that if you previously followed our guide on connecting a PS3 controller to your Raspberry Pi via Bluetooth, you will have to follow the next couple of steps. Otherwise, skip to step 2 of this section.

1. The reason for having to remove SIXAD is that it seems to conflict with the PlayStation 4 controllers and does not allow them to pair and function correctly.

AD

To uninstall the sixad package that we compiled and installed in our PS3 section we need to just run the following command on the Raspberry Pi.

Terminal \$

 Copy

```
sudo dpkg -r sixad
```

2. Now before we get to set up everything lets make sure our Raspberry Pi is entirely up to date by running the following two commands on it.

AD Terminal \$

 Copy

```
sudo apt update  
sudo apt upgrade
```

- 3.** Now let's begin the process of pairing our PS4 controller with our Raspberry Pi. For this guide, we will be using the Bluetooth command line tool.

Run the following command on your Raspberry Pi to start it up.

Terminal \$

 Copy

```
sudo bluetoothctl
```



- 4.** With the bluetoothctl tool now up we need to run the following two commands in it, this we enable the agent and set the agent to the default.

Terminal \$

 Copy

```
agent on  
default-agent
```

AD

5. Now that we have enabled the agent we can finally start scanning for devices. Switching scan on is crucial as it will show up our controller when we tell it to start syncing.

```
Terminal $
```

 Copy

```
scan on
```

AD

6. Now is the time to press both your “Share” button and PS button on your controller at the same time like shown in the diagram below. Hold these till the light on your controller starts flashing.



7. You should see something like below appear on your command line, once you see this, make a note of the MAC address.

The MAC address is the hexadecimal number separated by the ‘:’ symbol.

Output

```
[NEW] Device 00:01:6C:B4:06:7E Wireless Controller
```

In this case, the MAC Address is “**00:01:6C:B4:06:7E**”.

8. While your controller is still flashing, type in the following command. Make sure that you replace “**YOUR_MAC_ADDRESS**” with the MAC address that you got in **Step 8**. This command makes a connection with your PS4 controller.

If your controller stops flashing just press the “**Share**” and “**PS**” buttons again.

Terminal \$



```
connect YOUR_MAC_ADDRESS
```

9. Upon a successful connection, you should see the following appear in your command line.

AD



Output

Attempting to connect to 00:01:6C:B4:06:7E

[CHG] Device 00:01:6C:B4:06:7E Connected: yes

[CHG] Device 00:01:6C:B4:06:7E UUIDs: 00001124-0000-1000-8000-00805f9b34fb

[CHG] Device 00:01:6C:B4:06:7E UUIDs: 00001200-0000-1000-8000-00805f9b34fb

[CHG] Device 00:01:6C:B4:06:7E ServicesResolved: yes

[CHG] Device 00:01:6C:B4:06:7E Paired: yes

Connection successful

10. Now we need to add our MAC address to the trusted list so the PS4 controller can automatically connect to the Raspberry Pi.

Terminal \$

 Copy

```
trust YOUR_MAC_ADDRESS
```

11. With the PS4 controller now trusted by the Raspberry Pi's Bluetooth device, we can now quit out of the tool by pressing `CTRL + D` or typing in “`quit`”.

We can now proceed on to testing whether the controller is working.

AD

- 12.** Let's now install the joystick package, while this comes with some installations of Raspbian we will run the install command just in case. This package includes the applications we need to be able to test whether our controller is working correctly.

AD



**Ukrainian children need
your support.**

DONATE NOW

Run the following [Linux command](#) on your Raspberry Pi to install the software.

Terminal \$

Copy

```
sudo apt install joystick
```

- 13.** With the joystick package now installed to your Raspberry Pi you can now run the following command, this will test js0 and receive the values back from it.

Terminal \$

Copy

```
sudo jstest /dev/input/js0
```

- 14.** Now if values change when you move and press buttons on your controller, then you are ready to go with your PS4 controller, and you can finish the tutorial here.

However, if your controller refuses to work, you may have to try the alternative way and utilize the userspace driver, follow our guide in the next section below.

Using ds4drv to connect the PS4 Controller to the Raspberry Pi

AD

Only utilize this option if standard Bluetooth doesn't work at all. This method makes use of a userspace driver that talks with the controller and allow it to be understood by the Raspberry Pi.

You will need to of already paired your PS4 controller with your Raspberry Pi before you can continue with the following steps.

1. To begin, we need to install the packages we require for ds4drv. To install the necessary python3 components run the following command on your Raspberry Pi.

Terminal \$

 Copy

```
sudo apt install python3-dev python3-pip
```

2. With the required python packages now installed to the Raspberry Pi, we can use pip to grab ds4drv and the python libraries necessary for it. Run the following command to use pip3 to install it.

AD Terminal \$

 Copy

```
sudo pip3 install ds4drv
```

- 3.** Lets now setup the Raspberry Pi so non-root users can gain access to the joystick device that ds4drv creates, to do this run the following three commands on your Raspberry Pi. These commands download the ruleset then applies them.



**Ukrainian children need
your support.**

DONATE NOW

Terminal \$

 Copy

```
sudo wget https://raw.githubusercontent.com/chrippa/ds4drv/master/udev/50-ds4drv.rules  
sudo udevadm control --reload-rules  
sudo udevadm trigger
```

- 4.** Now while your PlayStation 4 controller is still paired to the Raspberry Pi, we can run the following command to see if ds4drv can read and interact with your controller.

Terminal \$

 Copy

```
sudo ds4drv --hidraw --led 000008
```

AD

5. Once you see text like below appear on your command line, that means that ds4drv can see and talk with your PS4 controller that is paired with the Raspberry Pi.

Output

```
[info][controller 1] Connected to Bluetooth Controller (00:01:6C:B4:06:7E hidraw5)
[info][hidraw] Scanning for devices
[info][controller 1] Battery: Fully charged
```

AD



**Ukrainian children need
your support.**

[DONATE NOW](#)

Now quit out of ds4drv by pressing **CTRL + C**.

6. We need to modify the bootup script now, so it automatically launches ds4drv at startup. To begin doing this run the following command on the Raspberry Pi.

Terminal \$

A small icon of a clipboard with a document, indicating a copy operation.

```
sudo nano /etc/rc.local
```

7. Now we need to add the following line to this file. This line will launch ds4drv when the Raspberry Pi boots up.

Find

AD Terminal \$

 Copy

exit 0

Add Above

Terminal \$

 Copy

/usr/local/bin/ds4drv --hidraw --led 000008 &

AD

Once you have made the changes, you can save the file by pressing `CTRL + X` then `Y` then hitting `ENTER`.

- 8.** You can now restart the Raspberry Pi to test whether it is starting up correctly. Do this by running the following command.

Terminal \$

 Copy

sudo reboot

- 9.** Once your Raspberry Pi has rebooted, you can now use the following command to test whether everything is working. If the numbers change then everything is working as it should.

AD Terminal \$

 Copy

```
sudo jstest /dev/input/js0
```

Conclusions

AD



**Ukrainian children need
your support.**

[DONATE NOW](#)

Hopefully, by the end of this guide, you have your Raspberry Pi up and running with your PlayStation controllers successfully connecting to it.

Be sure to also check out some of our other game tutorials for the [Pi such as Parsec streaming](#). If you have any feedback or need assistance with this guide, then feel free to drop a comment below.

Weekly Updates Straight To Your Inbox

Receive our Raspberry Pi projects, coding tutorials, Linux guides and more!

 Email*