# DeskPi Pro V3 Firmware update Manual

## Description

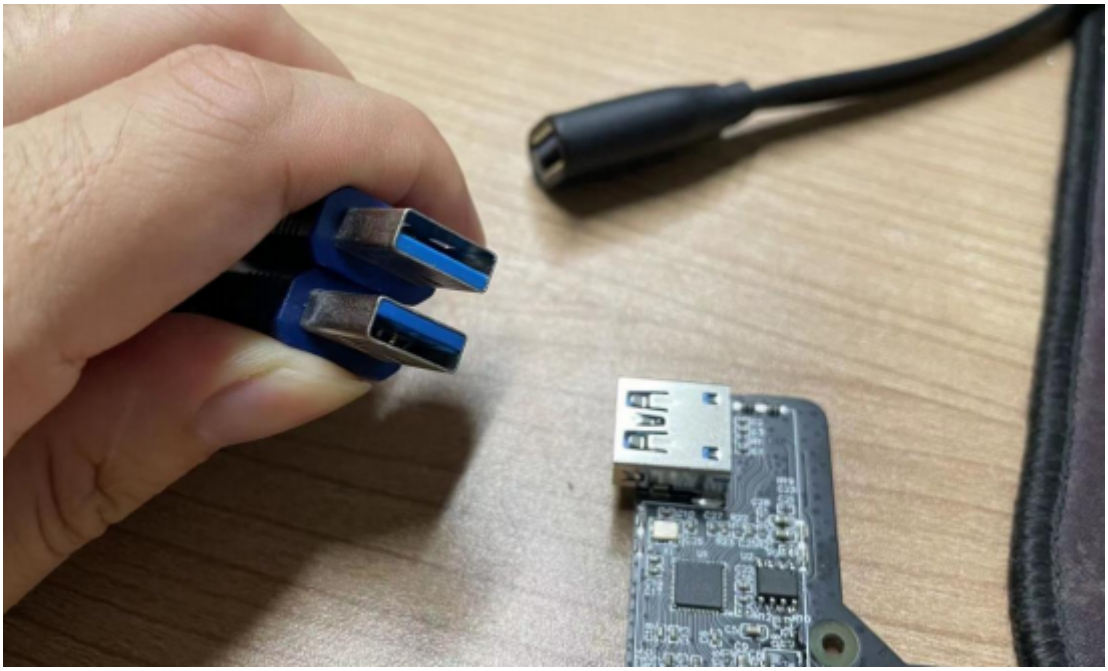
In order to better obtain the performance of the SSD drive, you can update the firmware and enable the trim function. Please follow the operation method below to update the firmware.

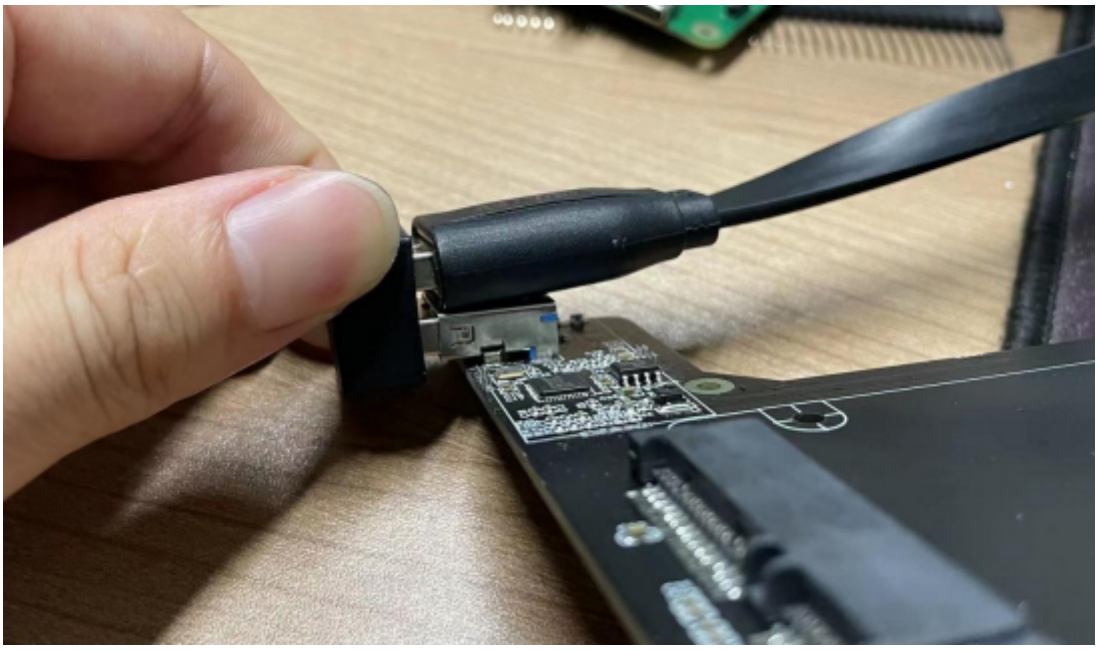
## What you need?

1. 1 x USB-A to USB-A Cable
2. 1 x Windows PC
3. 1 x DeskPi Pro V3 Daughter board
4. 1 x DeskPi Disk FW Flash Tool Pack

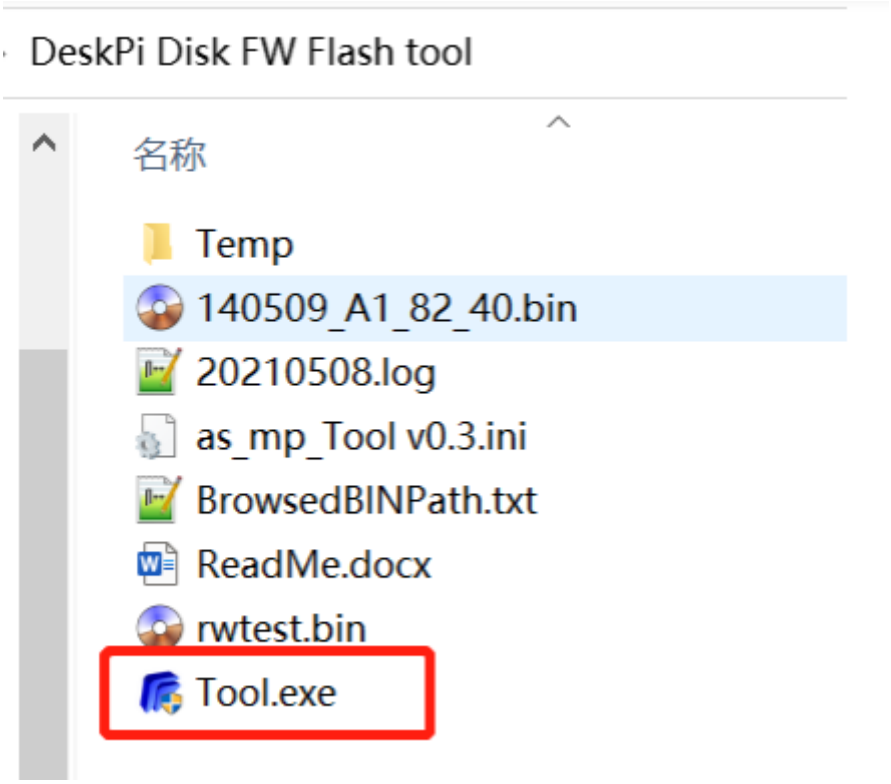
## Firmware update Method:

1. Use USB-A to USB-A Cable（2.0 or 3.0）connect the disk board to PC, or connect the plug directly to PC

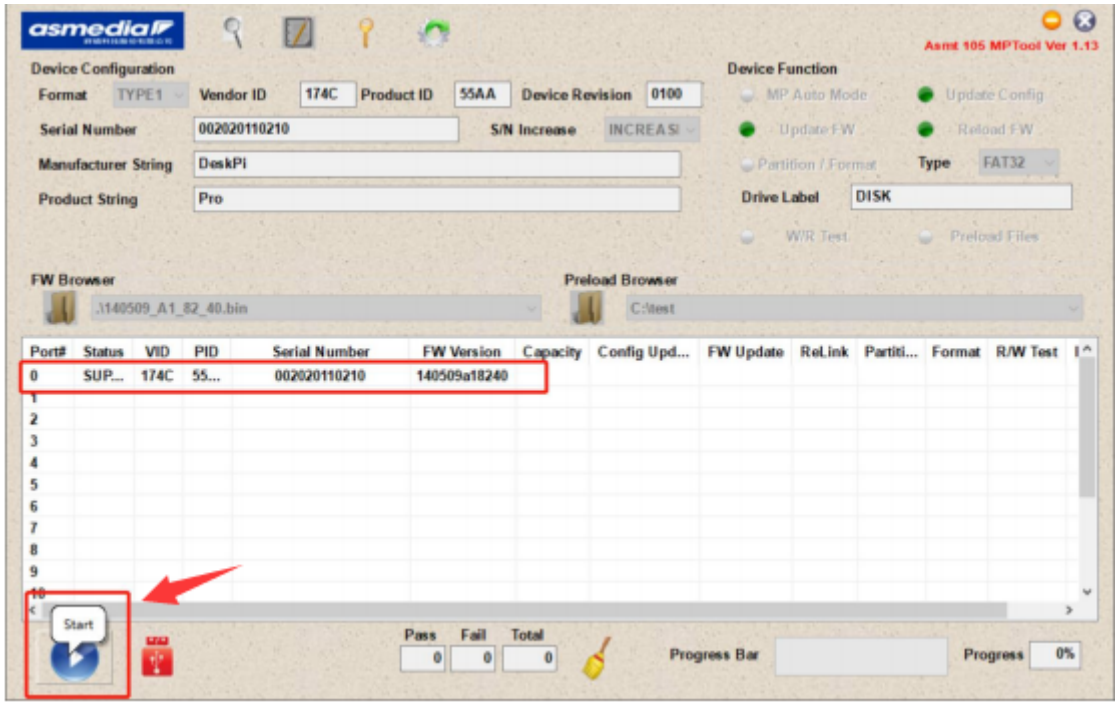




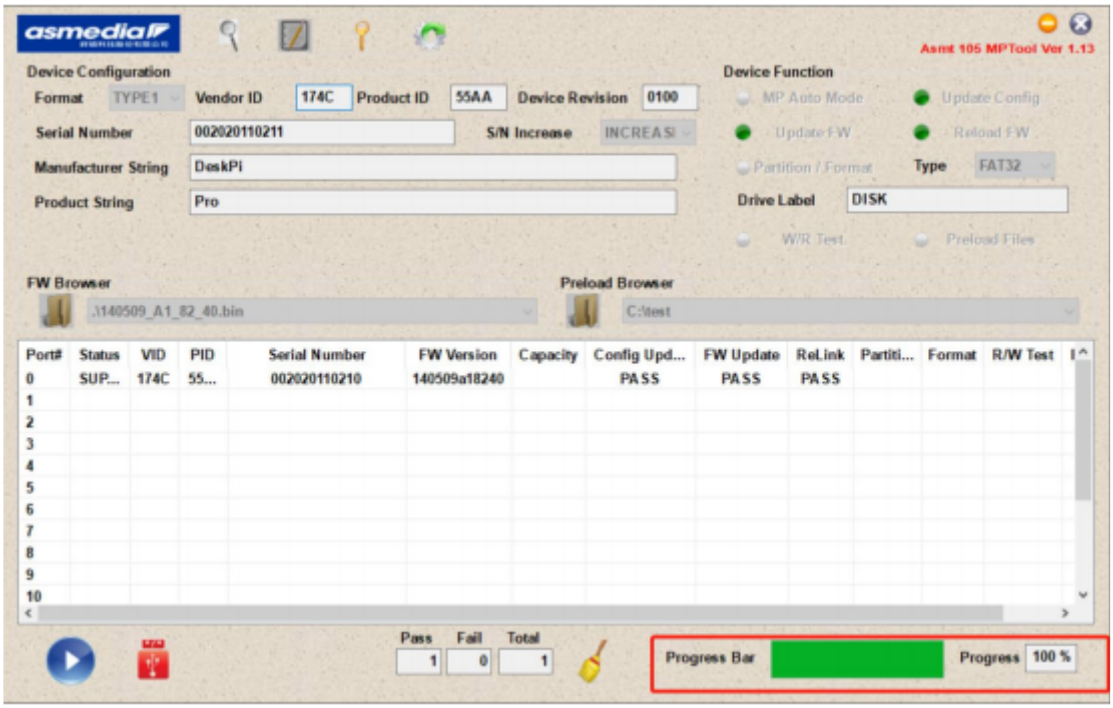
2. Unpack DeskPI Disk FW Flash Tool, and double click `Tool.exe`




3. Confirm the board has been found, click Start button. Wait for a moment。




When the progress bar is 100%, you can unplug the DeskPi Pro Daughter board and install the Raspberry Pi on the DeskPi Pro.




# How to enable trim on Raspberry Pi OS?

1. Check if SSD supports TRIM? Execute folloing command in terminal:

```
sudo fstrim -v /
```

If this reports back

```
fstrim: /: the discard operation is not supported, then TR
IM is not enabled.
```

You can also check with:

```
lsblk -D
```

If the `DISC-MAX` value is `0B`, then TRIM is not enabled.

2. Checking if the Firmware supports TRIM?

```
sudo apt-get -y install sg3-utils lsscsi
```

Run the following command and check the `Maximum unmap LBA count`:
by this command :

```
sg_vpd -p bl /dev/sda
```

it will show this:

```
sg_vpd -p bl /dev/sda
Block limits VPD page (SBC):
...
  Maximum unmap LBA count: 4194240
  Maximum unmap block descriptor count: 1
...
```

Take note of it, then run the following command and check the `Unmap command supported (LBPU)`:

```
sg_vpd -p lbpv /dev/sda
```

It will show:

```
Logical block provisioning VPD page (SBC):
  Unmap command supported (LBPU): 1
...
```

If the `Maximum unmap LBA count` is greater than `0`, and `Unmap command supported (LBPU)` is `1`, then the device firmware likely supports TRIM.

Warning: some device run fstrim command may corrupt the drive's firmware, to the point it won't mount and can't be formatted anymore. So make sure you have a backup of any important data before you try on a drive that might not actually support TRIM!

# Enable TRIM

Check on the current `provisioning_mode` for all drives attached to the Pi:

```
find /sys/ -name provisioning_mode -exec grep -H . {} + |
 sort
```

if you see something like this:

```
/sys/devices/platform/scb/fd500000.pcie/pci0000:00/0000:0
0:00.0/0000:01:00.0/usb2/2-1/2-1:1.0/host0/target0:0:0/0:
0:0:0/scsi_disk/0:0:0:0/provisioning_mode:full
```

Change the `provisioning_mode` from `full` to `unmap`; but if you have more than one drive attached, you need to confirm which drive you need to change. You can do that using lsscsi:

```
lsscsi
```

```
pi@raspberrypi:~ $ lsscsi
[0:0:0:0]    disk    KINGSTON  SA400M8120G     0     /dev/sda
pi@raspberrypi:~ $
```

Once you've confirmed which drive you need to change, change the value from `full` to `unmap` in the path that the find command returned:

```
echo unmap > /sys/devices/platform/scb/fd500000.pcie/pci00
00:00/0000:00:00.0/0000:01:00.0/usb2/2-1/2-1:1.0/host0/tar
get0:0:0/0:0:0:0/scsi_disk/0:0:0:0/provisioning_mode
```

Run the find command again to confirm the `provisioning_mode` is now `unmap`:

```
find /sys/ -name provisioning_mode -exec grep -H . {} + |
 sort
```

result:

```
/sys/devices/platform/scb/fd500000.pcie/pci0000:00/0000:0
0:00.0/0000:01:00.0/usb2/2-1/2-1:1.0/host0/target0:0:0/0:
0:0:0/scsi_disk/0:0:0:0/provisioning_mode:unmap
```

```
pi@raspberrypi:~ $ sudo find /sys/ -name provisioning_mode -exec grep -H . {} + | sort
/sys/devices/platform/scb/fd500000.pcie/pci0000:00/0000:00:00.0/0000:01:00.0/usb2/2-1/2-1:1.0/host0/target0:0:0/0:0:0:0/scsi_disk/0:
0:0:0/provisioning_mode:unmap
pi@raspberrypi:~ $
```

Now, you need to update the `discard_max_bytes` value for the drive, based on the `Maximum unmap LBA count` value you got from the `sg_vpd -p bl /dev/sda` command earlier, times the `Logical block length` value you get from the `sg_readcap -l /dev/sda` command. your values may be different:

```
root@raspberrypi:/home/pi# sg_vpd -p bl /dev/sda
Block limits VPD page (SBC):
  Write same non-zero (WSNZ): 0
  Maximum compare and write length: 0 blocks [Command
  Optimal transfer length granularity: 1 blocks
  Maximum transfer length: 65535 blocks
  Optimal transfer length: 65535 blocks
  Maximum prefetch transfer length: 65535 blocks
  Maximum unmap LBA count: 4194240
  Maximum unmap block descriptor count: 1
  Optimal unmap granularity: 1 blocks
```

```
root@raspberrypi:/home/pi# sg_readcap -l /dev/sda
Read Capacity results:
   Protection: prot_en=0, p_type=0, p_i_exponent=0
   Logical block provisioning: lbpme=0, lbprz=0
   Last LBA=234441647 (0xdf94baf)   Number of logical blocks=234441648
   Logical block length=512 bytes
   Logical blocks per physical block exponent=0
   Lowest aligned LBA=0
Hence:
   Device size: 120034123776 bytes, 114473.5 MiB, 120.03 GB
```

```
echo $((4194240*512))
```

result : 2147450880

Then write that value into the drive's `discard_max_bytes` setting.

```
echo 2147450880 > /sys/block/sda/queque/discard_max_bytes
```

Now, to confirm TRIM is enabled, run:

```
fstrim -v  /
```

```
root@raspberrypi:/home/pi# fstrim -v /
/: 103.5 GiB (111100010496 bytes) trimmed
root@raspberrypi:/home/pi#
```

it could take a few seconds

## Make it available after rebooting Pi.

These values will all be reset next time you reboot the Pi. To make the rules stick, you need to add a udev rule:

```
sudo nano /etc/udev/rules.d/10-trim.rules
```

And add the following in that file:

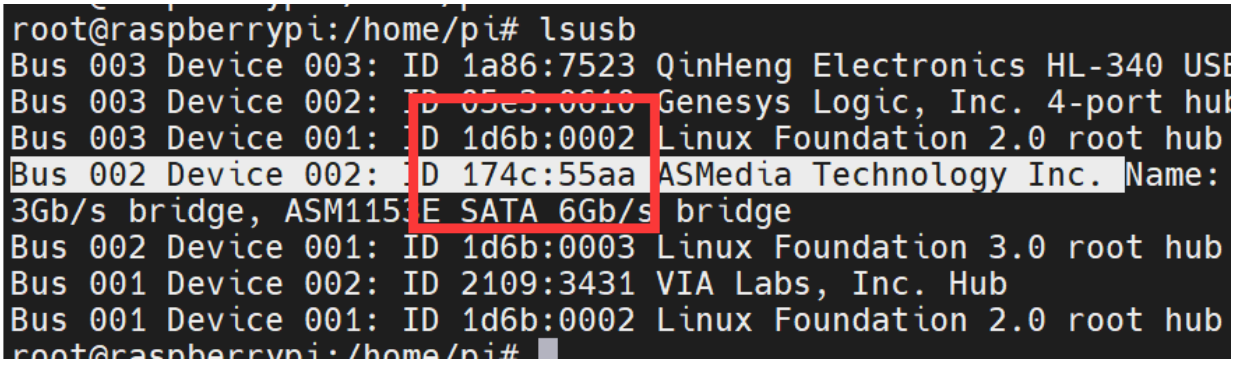
```
ACTION=="add|change", ATTRS{idVendor}=="174c", ATTRS{idPro
duct}=="55aa", SUBSYSTEM=="scsi_disk", ATTR{provisioning_m
ode}="unmap"
```

```
root@raspberrypi:/home/pi# cat /etc/udev/rules.d/10-trim.rules
ACTION=="add|change", ATTRS{idVendor}=="174c", ATTRS{idProduct}=="55aa", SUBSYSTEM=="scsi_disk", ATTR{provisioning_mode}="unmap"
root@raspberrypi:/home/pi#
```

you can get the `idVendor` and `idProduct` by using `lsusb` utility:

```
lsusb
```

```
root@raspberrypi:/home/pi# lsusb
Bus 003 Device 003: ID 1a86:7523 QinHeng Electronics HL-340 USB-Serial adapter
Bus 003 Device 002: ID 05e3:0610 Genesys Logic, Inc. 4-port hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 174c:55aa ASMedia Technology Inc. Name: ASM1051E SATA 6Gb/s bridge, ASM1053E SATA 6Gb/s bridge, ASM1153 SATA
3Gb/s bridge, ASM1153E SATA 6Gb/s bridge
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@raspberrypi:/home/pi#
```




And looking at the ‘ASMedia’ line, the vendor is the first part of the identifier (174c), and the product is the second part (55aa).

Make sure to save your `10-trim.rules` file, then `reboot` the Pi.

Try running `fstrim` again, and make sure it works:

```
sudo fstrim -v /
/: 102.3 GiB (109574424254 bytes) trimmed
```

The first time fstrim is run after a reboot, it will trim all the free space, which is why it gives such a large number. From that point on, the kernel will track changed blocks and trim only that data until the next boot.

# How to enable Automatic trimming

make sure the TRIM command is run automatically in the background is to enable the built-in `fstrim.timer`.

To do that, run the command:

```
sudo systemctl enable fstrim.timer
```

# How to diagnostic Disk speed:

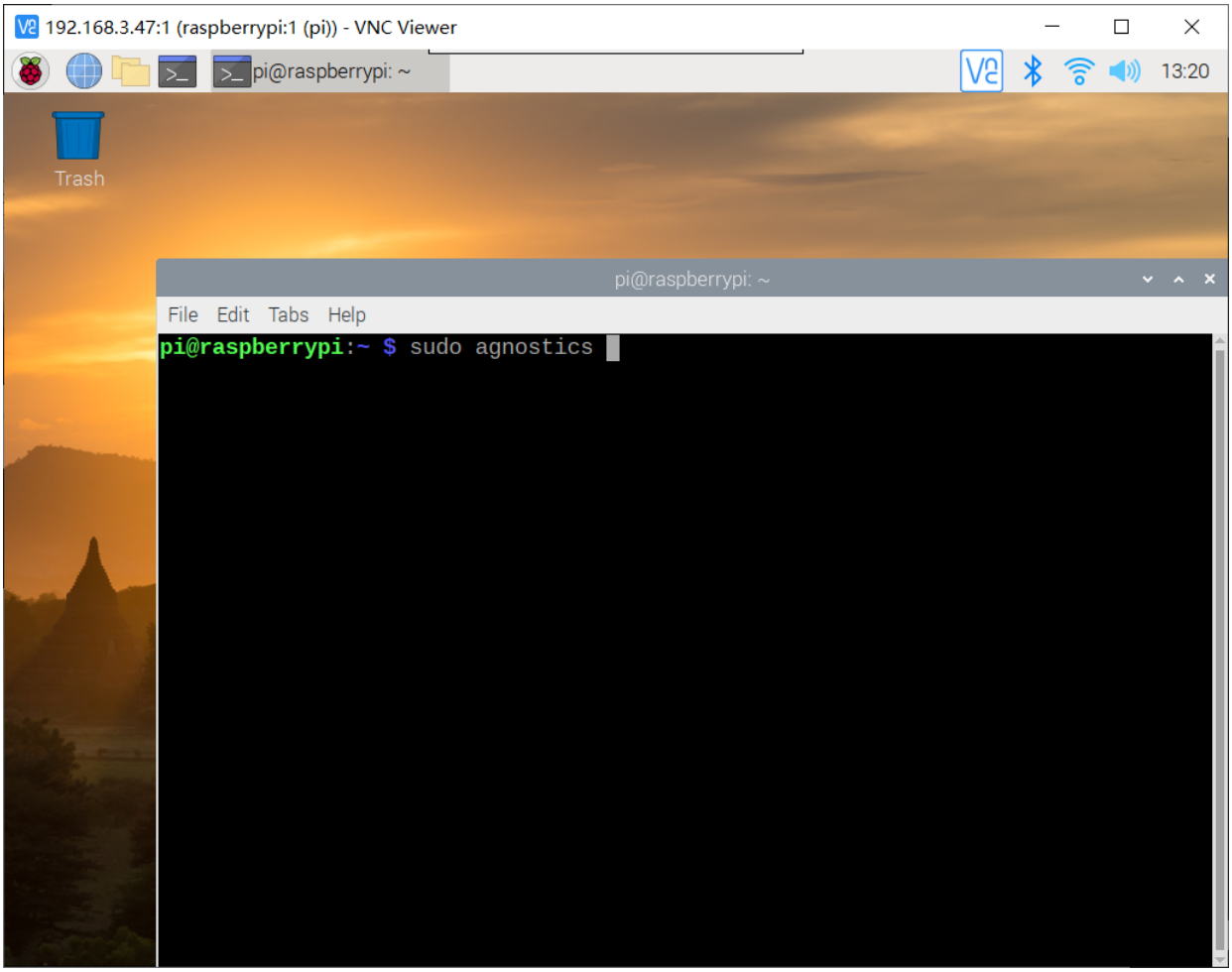
1. Install agnostic software

```
sudo apt-get -y install agnostic
```

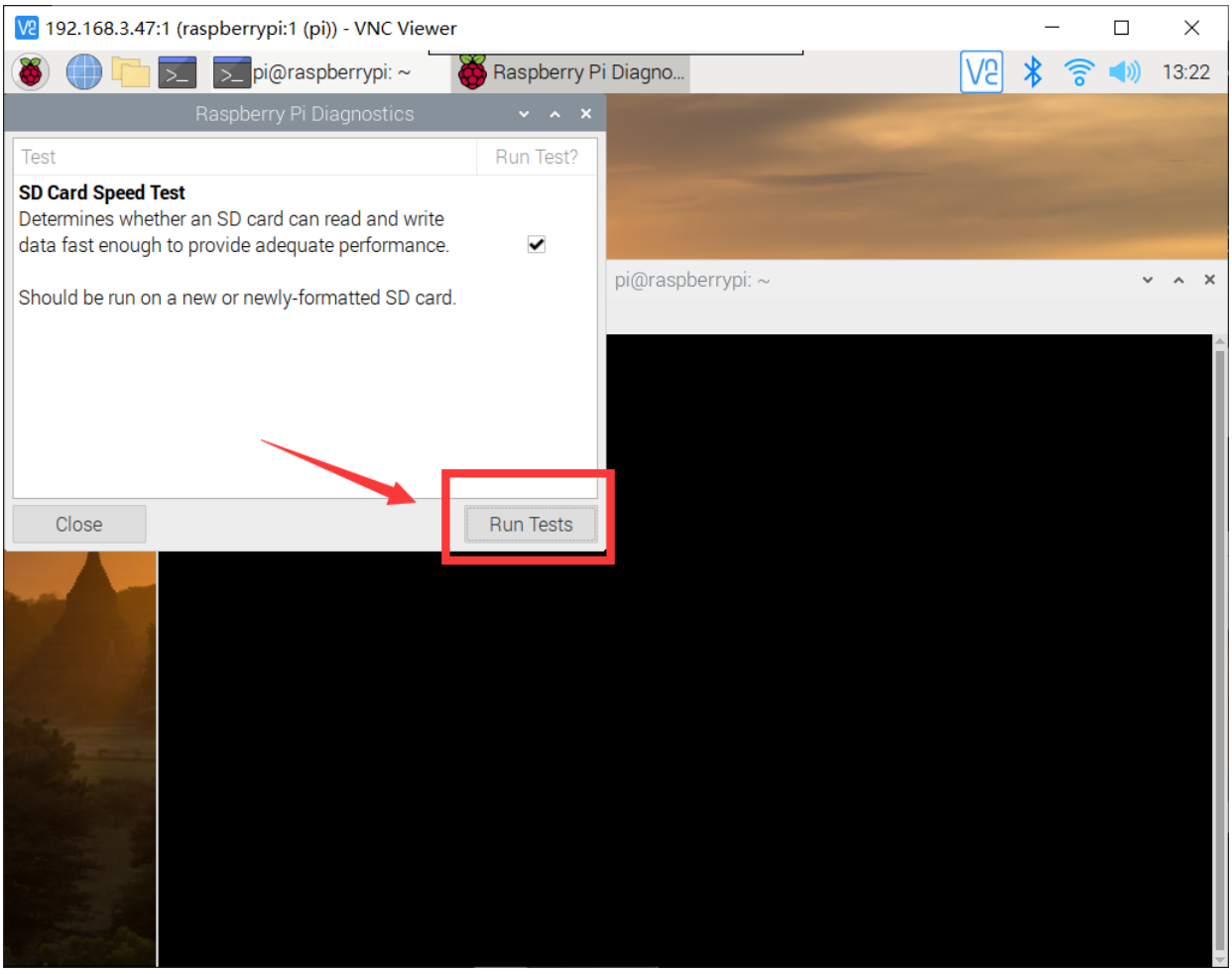
```
pi@raspberrypi:~ $ sudo apt-get -y install agnostics
Reading package lists... Done
Building dependency tree
Reading state information... Done
agnostics is already the newest version (0.9).
agnostics set to manually installed.
The following packages were automatically installed and are no longer required:
  gconf-service gconf2-common libgconf-2-4 lxplug-volume
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 161 not upgraded.
pi@raspberrypi:~ $
```

2. Execute following command in terminal on desktop

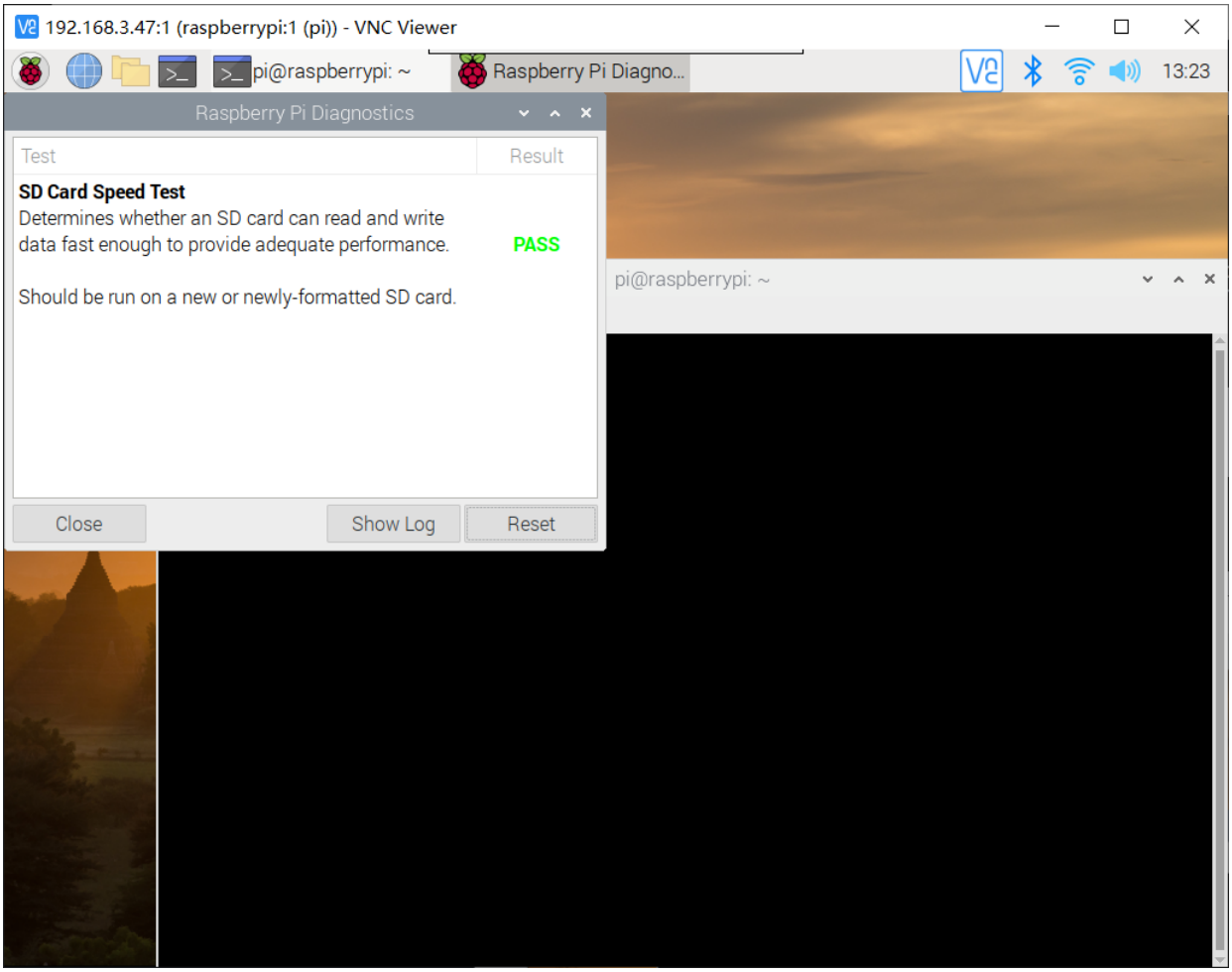
```
sudo agnostics
```



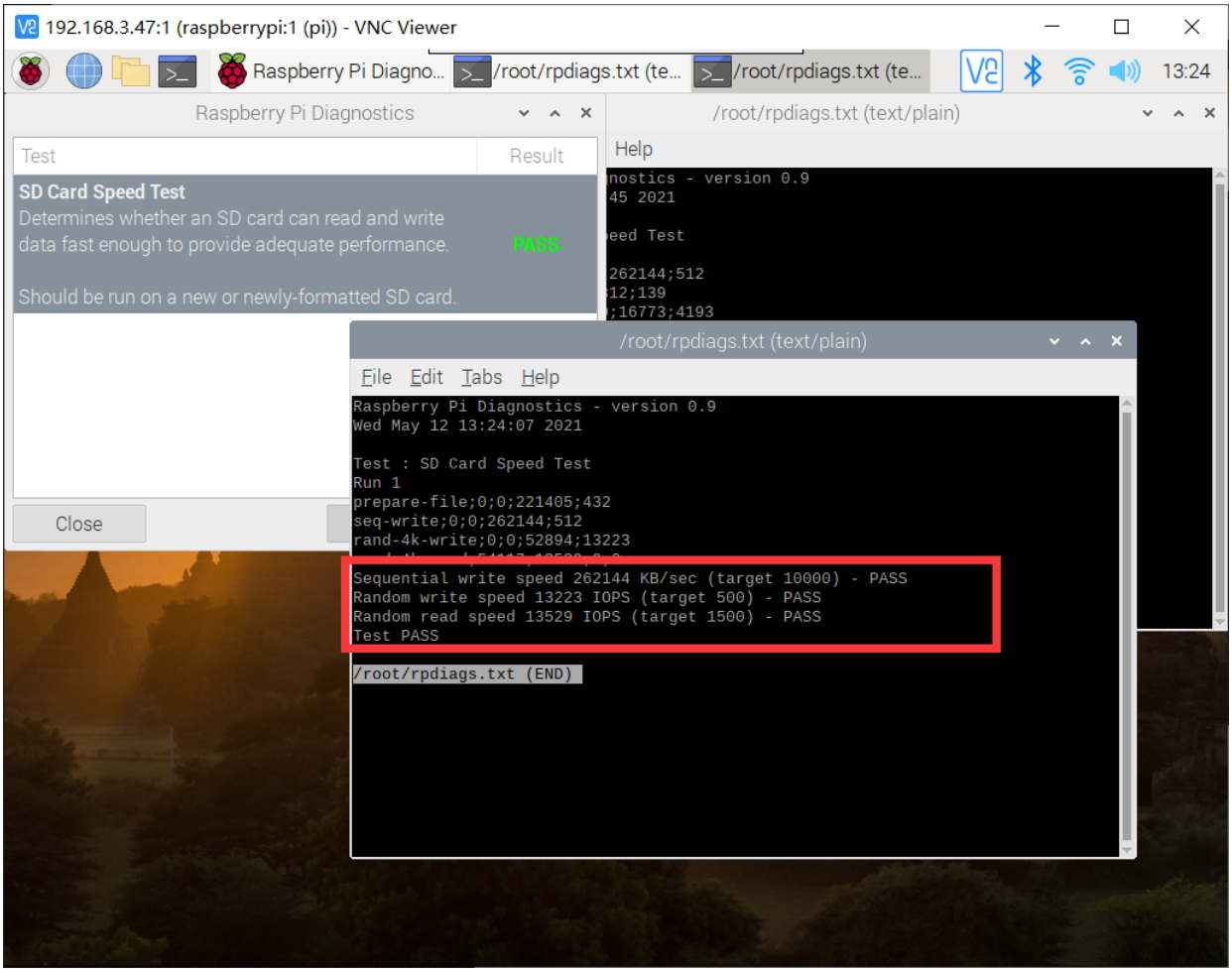

3. Press `Run tests`



4. After testing.Press `show log` bottom.




and you will get the result of your SSD speed testing.


192.168.3.47:1 (raspberrypi:1 (pi)) - VNC Viewer
Raspberry Pi Diagno...
/root/rpdiags.txt (te...
/root/rpdiags.txt (te...
13:24
Raspberry Pi Diagnostics
Test
Result
SD Card Speed Test
Determines whether an SD card can read and write data fast enough to provide adequate performance.
PASS
Should be run on a new or newly-formatted SD card.
Close
/root/rpdiags.txt (text/plain)
Help
/root/rpdiags.txt (text/plain)
File Edit Tabs Help
Raspberry Pi Diagnostics - version 0.9
Wed May 12 13:24:07 2021
Test : SD Card Speed Test
Run 1
prepare-file;0;0;221405;432
seq-write;0;0;262144;512
rand-4k-write;0;0;52894;13223
Sequential write speed 262144 KB/sec (target 10000) - PASS
Random write speed 13223 IOPS (target 500) - PASS
Random read speed 13529 IOPS (target 1500) - PASS
Test PASS
/root/rpdiags.txt (END)