



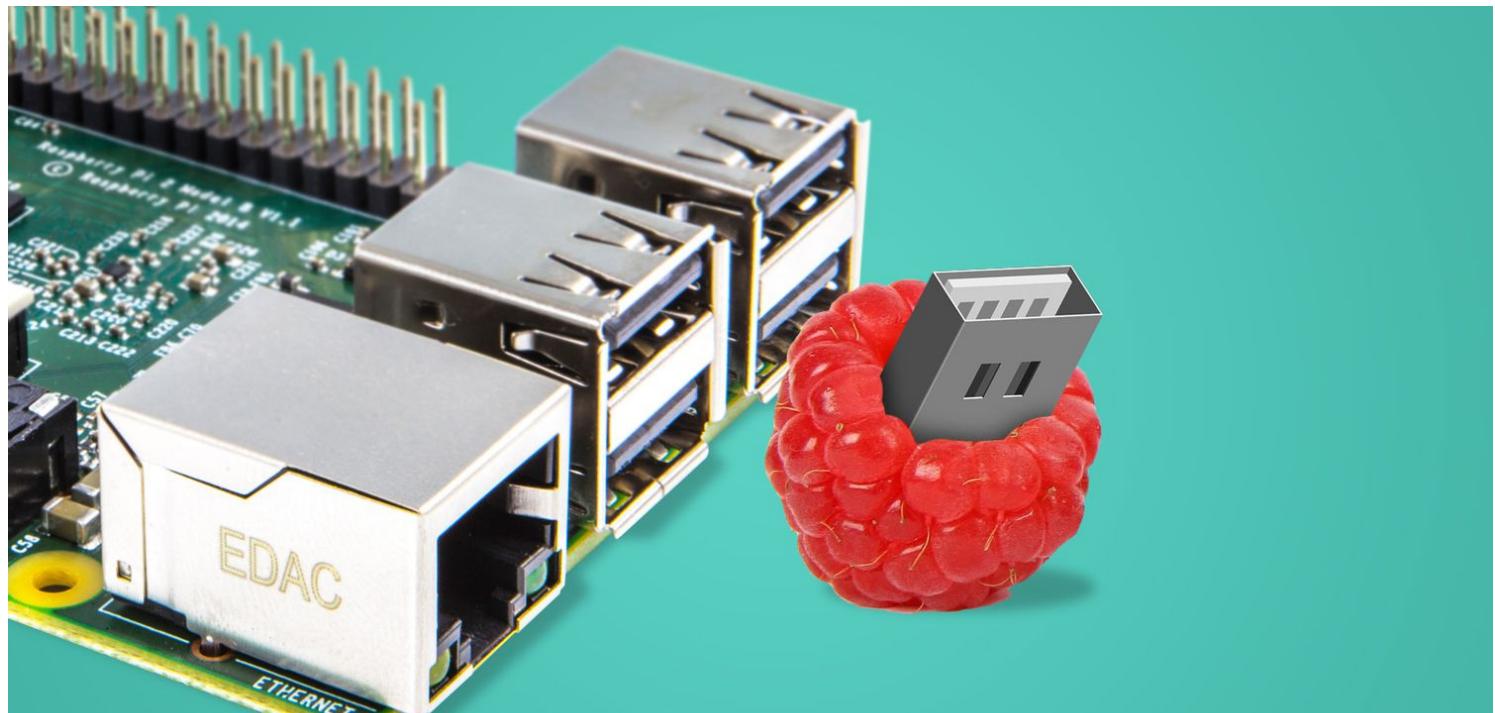
HOME > DIY

How to Make Raspberry Pi 3 Boot From USB

The Raspberry Pi is a versatile piece of kit, capable of a diverse range of tasks. But it has one glaring flaw: the inability to boot from USB. Until now, that is.

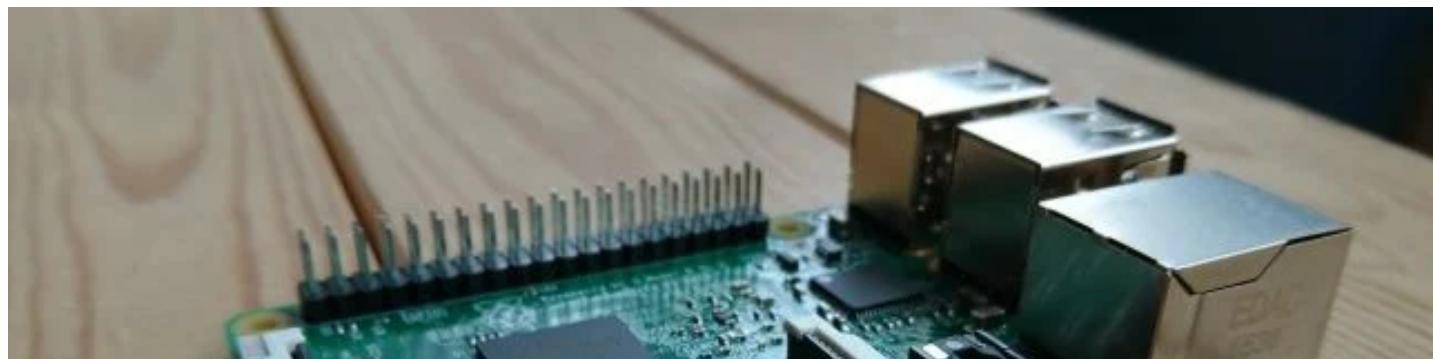
BY CHRISTIAN CAWLEY

PUBLISHED SEP 08, 2016



The Raspberry Pi is a great, versatile piece of kit, capable of projects as diverse as running a media center to use as a broadcast radio. But it has one glaring flaw: the inability to boot from USB.

Well, until now, that is.





If you're using a [**Raspberry Pi 3**](#), it is now possible to forego booting from microSD and instead boot the computer from a USB device. This might be a flash stick, an SSD with a USB adaptor, or even a full sized USB hard disk drive. This is a significant development, so let's take a look at how you can setup your Raspberry Pi 3 to boot from USB.

Get Started: Install Raspbian and Add New Files

It's best to start this project with a fresh copy of Raspbian, so download the [**latest version**](#) (we're using [**Raspbian Jessie**](#)) and [**install it in the usual way**](#). As soon as this is done, safely remove the card from your PC, insert it into the powered-down Raspberry Pi and boot, [**remote connecting over SSH**](#) as soon as it loads up.

Sign in (unless you've changed your [**default credentials**](#)) then run the following commands, which will replace the default

```
start.elf
```

and

```
bootcode.bin
```

files with freshly downloaded alternatives:

```
sudo apt-get update  
sudo BRANCH=next rpi-update
```

This update delivers the two files into the

```
/boot
```

directory. With the files downloaded, proceed to enable the USB boot mode with:

MAKEUSEOF VIDEO OF THE DAY

```
echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
```

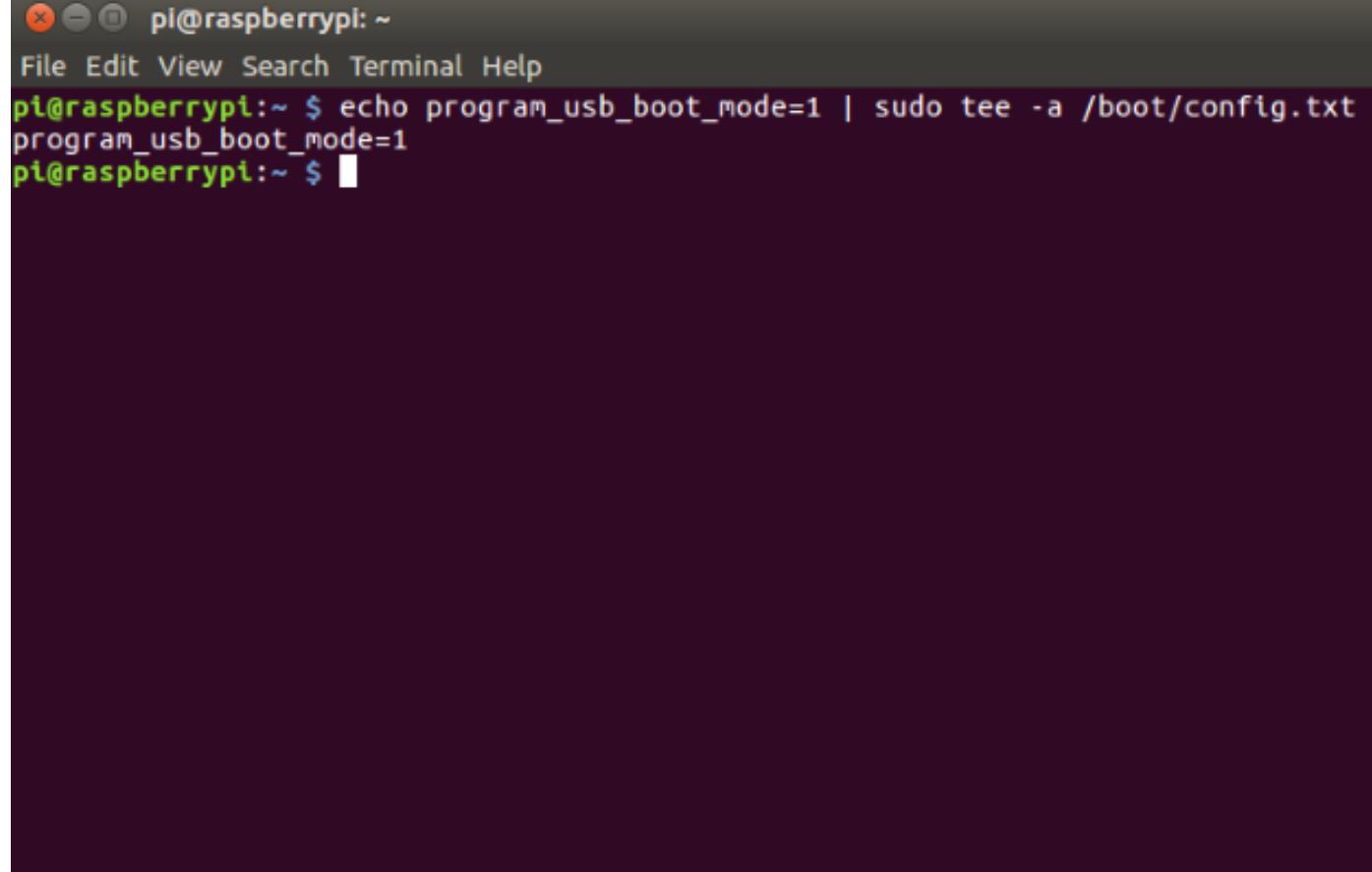
This command adds the

```
program_usb_boot_mode=1
```

instruction to the end of the

```
config.txt
```

file.



```
pi@raspberrypi: ~
File Edit View Search Terminal Help
pi@raspberrypi:~ $ echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
program_usb_boot_mode=1
pi@raspberrypi:~ $
```

You'll need to reboot the Pi once this is done.

Next step is to check that the OTP -- one-time programmable memory -- has been changed.
Check this with:

```
vcgencmd otp_dump | grep 17:
```

If the result is representative of the address

```
0x3020000a
```

(such as

```
17:3020000a
```

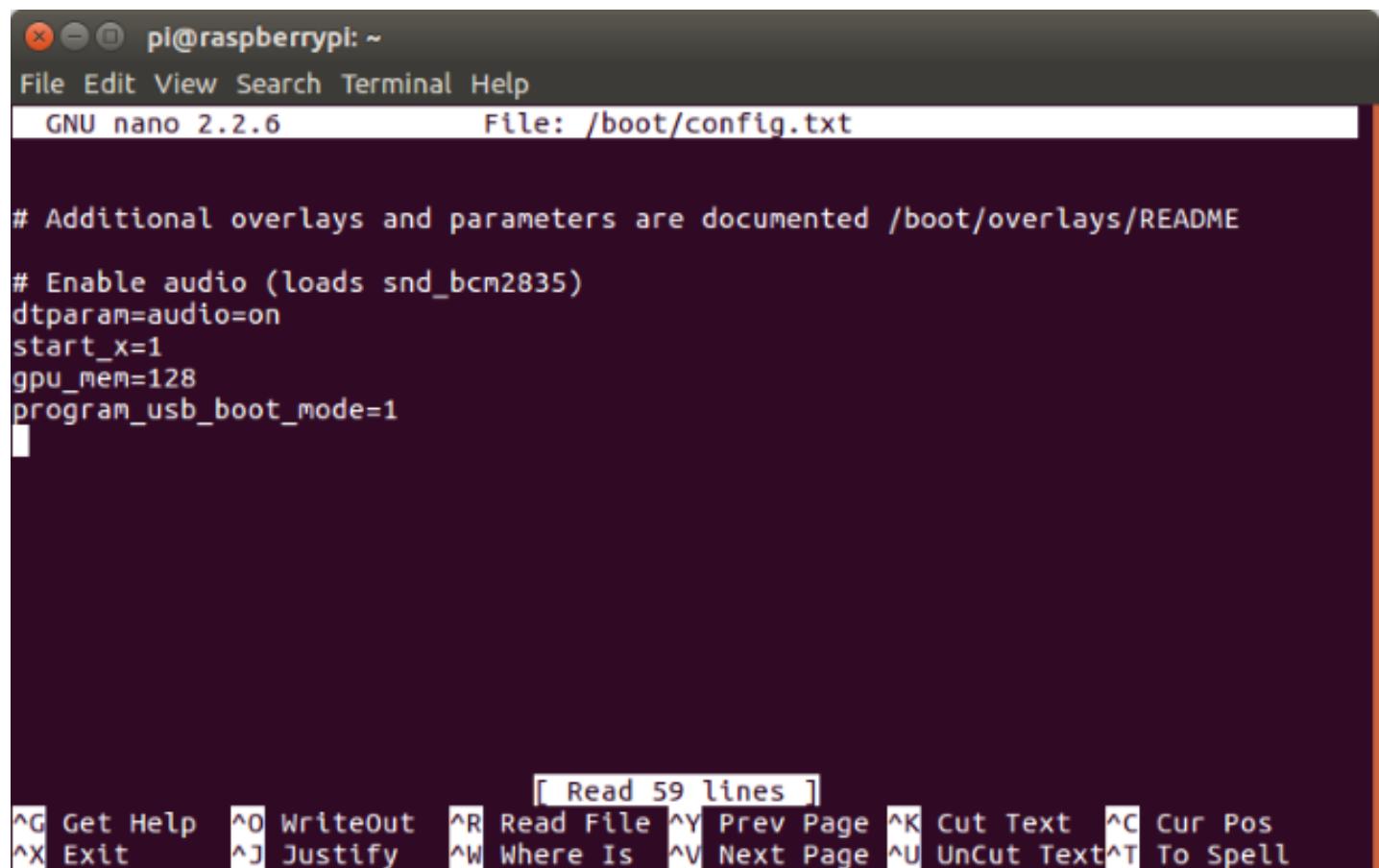
) then all is good so far. At this stage, should you wish to remove the

```
program_usb_boot_mode=1
```

line from the

```
config.txt
```

file, you can. The Pi is now USB boot-enabled, and you might wish to use the same microSD card in another Raspberry Pi 3, with the same image, so removing the line is a good idea.



The screenshot shows a terminal window titled "pi@raspberrypi: ~". The window title bar includes "File Edit View Search Terminal Help" and the status bar shows "GNU nano 2.2.6 File: /boot/config.txt". The main area of the terminal displays the contents of the /boot/config.txt file:

```
# Additional overlays and parameters are documented /boot/overlays/README
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
start_x=1
gpu_mem=128
program_usb_boot_mode=1
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for navigating and editing the file. The menu items include:

- Get Help (Alt+G)
- WriteOut (Alt+O)
- Read File (Alt+R)
- Prev Page (Alt+Y)
- Cut Text (Alt+K)
- Cur Pos (Alt+C)
- Exit (Alt+X)
- Justify (Alt+J)
- Where Is (Alt+W)
- Next Page (Alt+V)
- UnCut Text (Alt+U)
- To Spell (Alt+T)

A status message "[Read 59 lines]" is displayed above the menu bar.

This is easily done by editing

```
config.txt
```

in nano:

```
sudo nano /boot/config.txt
```

Delete or comment out the corresponding line (with a preceding #).

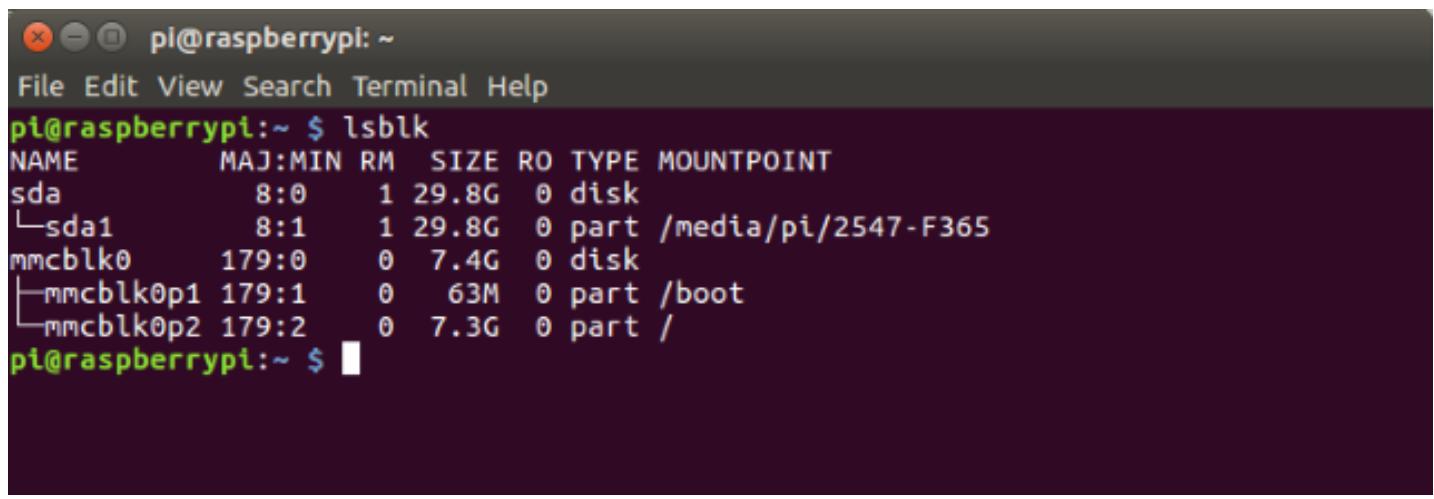
Prepare Your USB Boot Device

Next, connect a formatted (or ready-to-be-deleted) USB stick into a spare port on your Raspberry Pi 3. With this inserted, we'll proceed to copy the OS across.

Begin by identifying your USB stick, with the

```
lsblk
```

command.



```
pi@raspberrypi: ~
File Edit View Search Terminal Help
pi@raspberrypi:~ $ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    1 29.8G  0 disk 
└─sda1     8:1    1 29.8G  0 part /media/pi/2547-F365
mmcblk0   179:0   0  7.4G  0 disk 
└─mmcblk0p1 179:1   0   63M  0 part /boot
└─mmcblk0p2 179:2   0   7.3G  0 part /
pi@raspberrypi:~ $
```

In this example, the SD card is

```
mmcblk0
```

while the USB stick is

```
sda
```

(it's formatted partition is

```
sda1
```

). If you have other USB storage devices connected the USB stick might be sdb, sdc, etc. With the name of your USB stick established, unmount the disk and use the parted tool to create a 100 MB partition (FAT32) and a Linux partition:

```
sudo umount /dev/sda  
sudo parted /dev/sda
```

At the (parted) prompt, enter:

```
mktabe msdos
```

You might be informed that the disk is otherwise engaged. If so, select **Ignore**, then note the warning instructing you that the data on the disk will be destroyed. As explained earlier, this should be a disk that you're happy to delete or format, so agree to this.

If you run into any problems here, you might need to switch to the desktop (either manually, or over VNC) and confirm the disk is unmounted, before entering the mktabe msdos command in a windowed command line.

Proceed in parted with the following:

```
mkpart primary fat32 0% 100M  
mkpart primary ext4 100M 100%  
print
```

This will output some information concerning disk and the new partitions. Proceed to exit parted with Ctrl + C, before creating the boot filesystem, and the root filesystem:

```
sudo mkfs.vfat -n BOOT -F 32 /dev/sda1  
sudo mkfs.ext4 /dev/sda2
```

You then need to mount the target filesystems, before copying your current Raspbian OS to the USB device.

```
sudo mkdir /mnt/target  
sudo mount /dev/sda2 /mnt/target/  
sudo mkdir /mnt/target/boot  
sudo mount /dev/sda1 /mnt/target/boot/  
sudo apt-get update; sudo apt-get install rsync  
sudo rsync -ax --progress / /boot /mnt/target
```

That last one is the final command that copies everything over, and so will take a while to complete. Time to make a coffee!



| | | | | | |
|-------------------|-----------|------|-----------|---------|-----------------------------|
| boot/bootcode.bin | 50,844 | 100% | 52.49kB/s | 0:00:00 | (xfr#251, ir-chk=1020/1315) |
| boot/cmdline.txt | 136 | 100% | 0.14kB/s | 0:00:00 | (xfr#252, ir-chk=1019/1315) |
| boot/config.txt | 1,657 | 100% | 1.71kB/s | 0:00:00 | (xfr#253, ir-chk=1018/1315) |
| boot/fixup.dat | 6,493 | 100% | 6.69kB/s | 0:00:00 | (xfr#254, ir-chk=1017/1315) |
| boot/fixup_cd.dat | 2,517 | 100% | 2.59kB/s | 0:00:00 | (xfr#255, ir-chk=1016/1315) |
| boot/fixup_db.dat | 9,725 | 100% | 10.00kB/s | 0:00:00 | (xfr#256, ir-chk=1015/1315) |
| boot/fixup_x.dat | 9,725 | 100% | 9.99kB/s | 0:00:00 | (xfr#257, ir-chk=1014/1315) |
| boot/issue.txt | 145 | 100% | 0.15kB/s | 0:00:00 | (xfr#258, ir-chk=1013/1315) |
| boot/kernel.img | 4,122,848 | 100% | 3.01MB/s | 0:00:01 | (xfr#259, ir-chk=1012/1315) |
| boot/kernel7.img | 4,222,568 | 100% | 5.90MB/s | 0:00:00 | (xfr#260, ir-chk=1011/1315) |
| boot/start.elf | 2,758,008 | 100% | 2.79MB/s | 0:00:00 | (xfr#261, ir-chk=1010/1315) |
| boot/start_cd.elf | 32,768 | 5% | 33.20kB/s | 0:00:17 | ■ |

Next, you need to refresh the SSH host keys, to maintain the connection with the reconfigured Raspberry Pi after an imminent reboot:

```
cd /mnt/target
sudo mount --bind /dev dev
sudo mount --bind /sys sys
sudo mount --bind /proc proc
sudo chroot /mnt/target
rm /etc/ssh/ssh_host*
dpkg-reconfigure openssh-server
exit
sudo umount dev
sudo umount sys
sudo umount proc
```

```
pi@raspberrypi: /mnt/target
File Edit View Search Terminal Help
pi@raspberrypi:~ $ cd /mnt/target
pi@raspberrypi:/mnt/target $ sudo mount --bind /dev dev
pi@raspberrypi:/mnt/target $ sudo mount --bind /sys sys
pi@raspberrypi:/mnt/target $ sudo mount --bind /proc proc
pi@raspberrypi:/mnt/target $ sudo chroot /mnt/target
root@raspberrypi:# rm /etc/ssh/ssh_host*
root@raspberrypi:# dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
2048 11:72:0f:b2:70:a9:18:69:42:ce:de:cb:51:83:d1:84 /etc/ssh/ssh_host_rsa_key.p
ub (RSA)
Creating SSH2 DSA key; this may take some time ...■
```

Note that after sudo chroot (the fifth command above) you're switching to root, so the user will change from **pi@raspberrypi** to **root@raspberrypi** until you enter exit on line 8.

Prepare for Rebooting from USB!

Just a few more things to sort out before your Raspberry Pi is ready to boot from USB. We need to edit

```
cmdline.txt
```

again from the command line with:

```
sudo sed -i "s,root=/dev/mmcblk0p2,root=/dev/sda2," /mnt/target/boot/cmdline.txt
```

Similarly, the following change needs to be made to fstab:

```
sudo sed -i "s,/dev/mmcblk0p,/dev/sda," /mnt/target/etc/fstab
```

You're then ready to unmount the filesystems before shutting down the Pi:

```
cd ~  
sudo umount /mnt/target/boot  
sudo umount /mnt/target  
sudo poweroff
```

Note that this uses the new

```
poweroff
```

command as an alternative to

```
shutdown
```

When the Pi has shutdown, disconnect the power supply before removing the SD card. Next, reconnect the power supply -- your Raspberry Pi should now be booting from the USB device! And for more help with your Pi, check out [**how to set up Wi-Fi and Bluetooth on Raspberry Pi**](#) **3.**

Ready to [**give Raspberry Pi 4 a try**](#)? Check out its features and specs.



[6 Ways Your Email Address Can Be Exploited by Scammers](#)

READ NEXT >



RELATED TOPICS

ABOUT THE AUTHOR



Christian Cawley

(1578 Articles Published)



Deputy Editor for Security, Linux, DIY, Programming, and Tech Explained, and Really Useful Podcast producer, with extensive experience in desktop and software support. A contributor to Linux Format magazine, Christian is a...

More From Christian Cawley →

SUBSCRIBE TO OUR NEWSLETTER

Join our newsletter for tech tips, reviews, free ebooks, and exclusive deals!

[CLICK HERE TO SUBSCRIBE](#)

ON THE WIRE

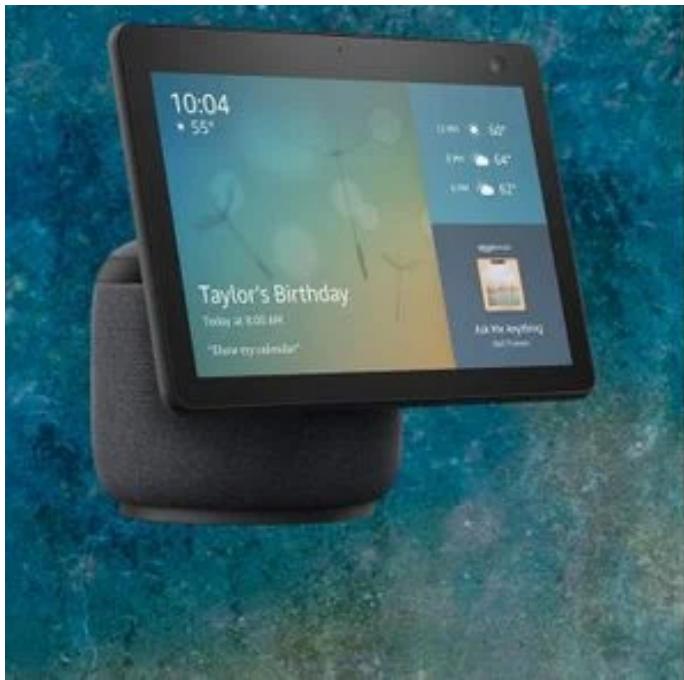


How to Disable USB Ports to Prevent Malware Infection

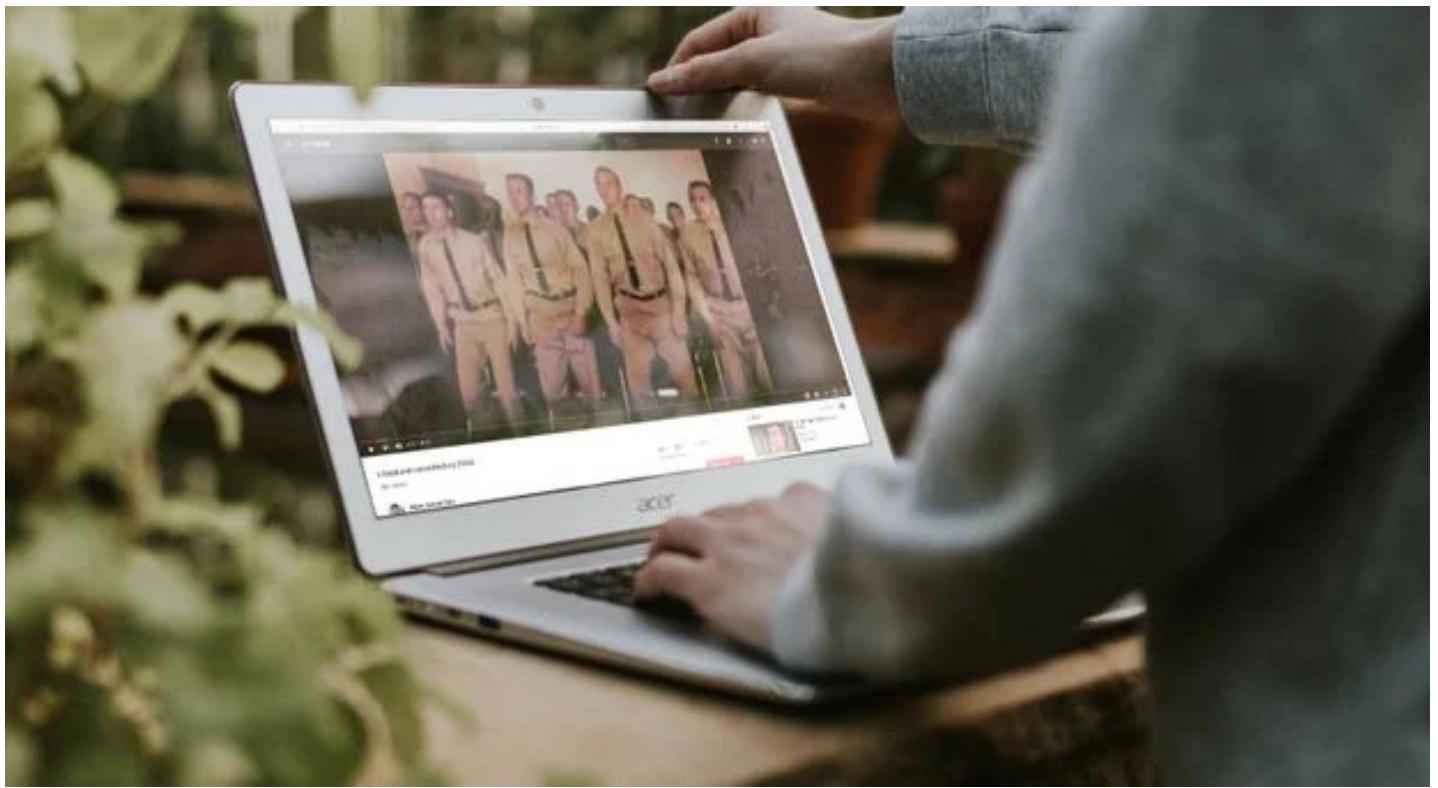


How Samsung Is Bringing Some S22 Features to Older Phones





Amazon Echo Show 10 vs. Echo Show 15: Which One Is Right for You?



The Best Free Movie Streaming Sites



Xbox Series X vs. Gaming PC: How Do They Compare?



How to Manually Update Microsoft Defender

READ NEXT



What Is an M.2 SSD? The Pros, Cons, and How to Install One



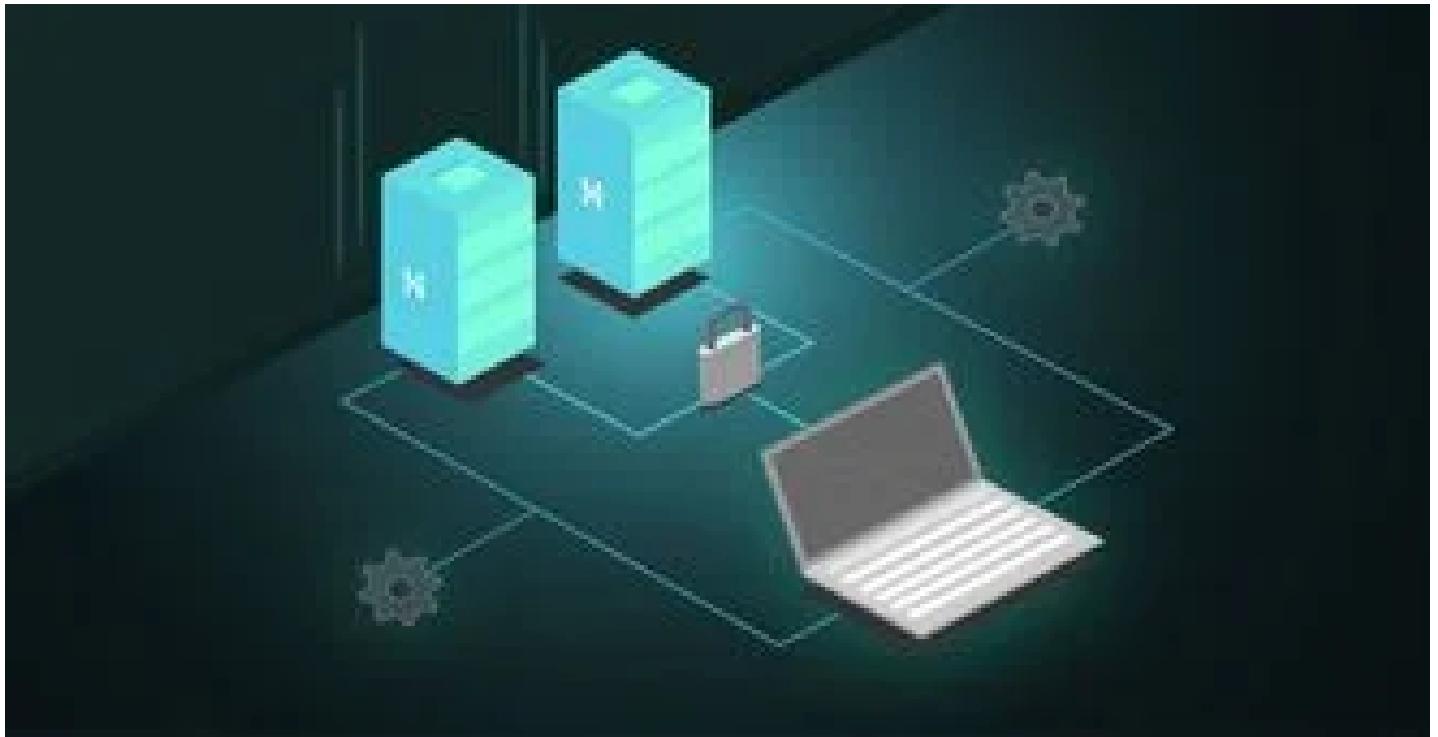
The Complete Guide to Windows 10 Privacy Settings



What Is an Electric Vehicle and How Does It Work?



How to Use Your Google Home as a Speaker



A Guide to MySQL for Beginners



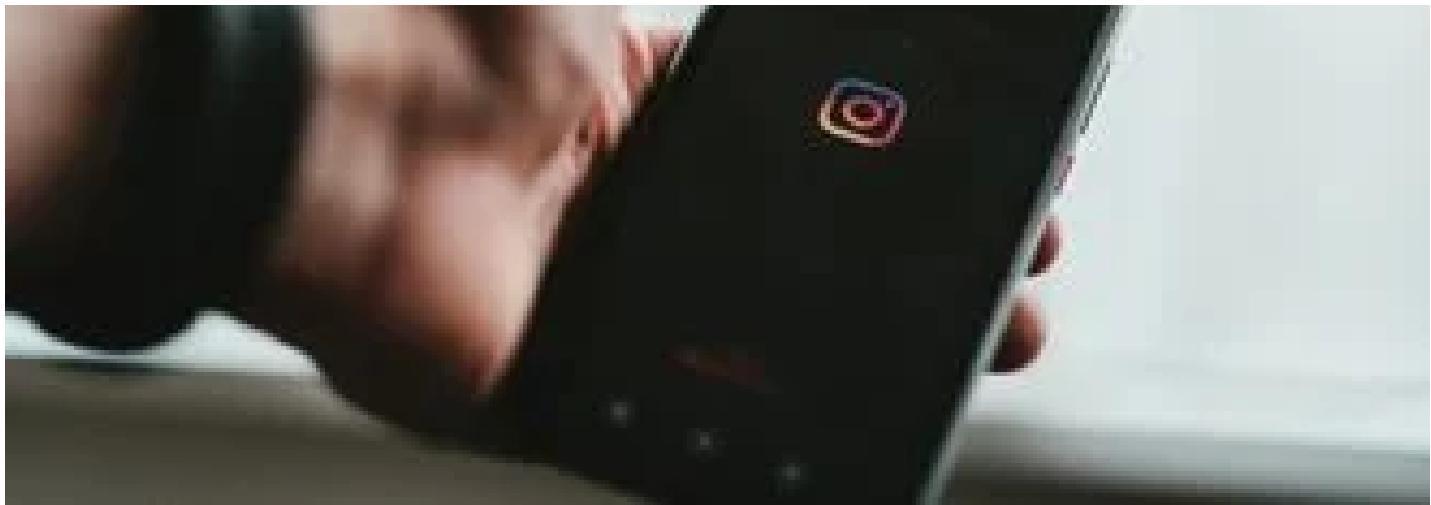


How UserSearch Can Help You Enjoy Better Dates



How to Customize Your Asana Homepage





10 Tricks to Get More Real Instagram Followers

 Computer on desk with error about low video memory showing

How to Increase Dedicated Video RAM (VRAM) in Windows 10 and 11

[Write For Us](#) [Home](#) [Contact Us](#) [Terms](#) [Privacy](#) [Copyright](#) [About Us](#) [Fact Checking Policy](#)

[Corrections Policy](#) [Ethics Policy](#) [Ownership Policy](#) [Partnership Disclaimer](#)

Copyright © 2022www.makeuseof.com