Part 03
-
Prepare An SD Card For Raspberry Pi

Version: 2020-03-16

What you need to start

To set up your Raspberry Pi you will need:

	Item	Minimum recommended specification & notes
1	SD card	Minimum size 4Gb; class 4 (the *class* indicates how fast the card is). We recommend using branded SD cards as they are more reliable.
2a	HDMI to HDMI / DVI lead	HDMI to HDMI lead (for HD TVs and monitors with HDMI input). OR HDMI to DVI lead (for monitors with DVI input).
2b	RCA video lead	A standard RCA composite video lead to connect to your analogue display if you are not using the HDMI output.
3	Keyboard and mouse	Any standard USB keyboard and mouse should work. Keyboards or mice that take a lot of power from the USB ports, however, may need a powered USB hub. This may include some wireless devices.
4	Ethernet (network) cable	Networking is optional, although it makes updating and getting new software for your Raspberry Pi much easier.
5	Power adapter	A good quality, micro USB power supply that can provide at least the power your Raspberry Pi model requires (see step 0) If your supply provides less than 5V then your Raspberry Pi may not work at all, or it may behave erratically. Be wary of very cheap chargers: some are not what they claim to be. It does not matter if your supply is rated more than what your Raspberry PI requires
6	Audio lead [optional]	If you are using HDMI then you will get digital audio via this. If you are using the analogue RCA connection, stereo audio is available from the 3.5mm jack next to the RCA connector.

Know your leads:

HDMI connector

HDMI to DVI lead

RCA composite video connector

Preparing your SD card for the Raspberry Pi with NOOBS

In order to use your Raspberry Pi, you will need to install an Operating System (OS) onto an SD card. An Operating System is the set of basic programs and utilities that allow your computer to run. Eg. Windows on a PC or OSX on a Mac.

The following instructions will guide you through installing an OS on your SD card that will allow you to easily install different OS's and to recover your card if you break it.

1. Insert an SD card that is sized between 4GB and 32 GB, into your computer

2. Format the SD card so that the Pi can read it

   A. Windows
   - Download the SD Association's Formatting Tool from `https://www.sdcard.org/downloads/formatter_4/eula_windows/`
     Note: *The builtin Windows formatting tool will only format the first partition that Windows can read not the entire disk. For this reason we advise using the official SD Card Association Formatting Tool.*
   - Install and run the Formatting Tool on your machine
   - Set "`FORMAT SIZE ADJUSTMENT`" option to "`ON`" in the "`Options`" menu
   - Check that the SD card you inserted matches the one selected by the Tool
   - Click the "`Format`" button

   B. Mac
   - Download the SD Association's Formatting Tool from `https://www.sdcard.org/downloads/formatter_4/eula_mac/`
   - Install and run the Formatting Tool on your machine
   - Select "`Overwrite Format`"
   - Check that the SD card you inserted matches the one selected by the Tool
   - Click the "`Format`" button

   C. Linux
   - We recommend using gparted (or the command line version parted)
   - Format the entire disk as FAT

3. Download the New Out Of Box Software (NOOBS) from `https://downloads.raspberrypi.org/noobs`

4. Unzip the downloaded file

   A. Windows
   Right-click on the file and choose "`Extract all`"

   B. Mac
   Double-tap on the file

   C. Linux
   Run `unzip [downloaded filename]`

5. Copy the extracted files onto the SD card that you just formatted

6. Insert the SD card into your Pi and connect the power supply. Your Pi will now boot into NOOBS and should display a list of operating systems that you can choose to install. If your display remains blank, you should select the correct output mode for your display by pressing one of the following number keys on your keyboard

   1 = HDMI mode : this is the default display mode.
   2 = HDMI safe mode **:** select this mode if you are using the HDMI connector and cannot see anything on screen when the Pi has booted.
   3 = Composite PAL mode : select either this mode or composite NTSC mode if you are using the composite RCA video connector
   4 = Composite NTSC mode

Install Raspberry Pi OS on SD card

Raspbian is a version of Debian Linux specifically configured to run on the Raspberry Pi and is recommended by the Raspberry Pi Foundation as the operating system to install.

1. Download the latest Raspbian OS available for free from the Raspberry Pi website. https://downloads.raspberrypi.org/raspbian_latest

2. Instaling is on SD card

   A. Windows

   Once you have the ZIP file downloaded to your computer, unarchive it using eg. 7zip (https://www.7-zip.org/). There will be a single .img file inside. This is the disk image you will flash to the Raspberry Pi's SD card. To install Raspbian, you will need an SD card that has at least 4 GB of space or more. Note that the cheap 16 GB Class 10 SD card works great on the Raspberry Pi, and gives you plenty of room to add media and other programs once Raspbian is installed.

   Insert the SD card into your SD card reader and check which drive letter was assigned. You can easily see the drive letter, such as G:, by looking in the left column of Windows Explorer. You can use the SD card slot if you have one, or a cheap SD adapter in a USB port.

   Download Rufus utility from https://rufus.ie/ . Install the tool and run it. You may need to run this as administrator. Right-click on the file, and select Run as administrator.
   Select the drive letter of the SD card in the device box. Be careful to select the correct drive. If you get the wrong one you can destroy the data on your computer's hard disk!
   If you are using an SD card slot in your computer and can't see the drive in Rufus, try using an external SD adapter.
   Select the image file .img you extracted earlier.
   Click Write and wait for the write to complete.
   Exit the imager and eject the SD card.

   An alternative is Etcher (https://www.balena.io/etcher/)

   B. Mac

   https://www.raspberrypi.org/documentation/installation/installing-images/mac.md

   C. Linux

   https://www.raspberrypi.org/documentation/installation/installing-images/linux.md

Automate making SD card with Raspbian OS installed on it

Here is a Windows CMD script that automates the writing of an SD Card

```
@echo off
cls
setlocal enabledelayedexpansion
:: set some vars
:: location/path of Rufus.exe
:: rufus.exe can be downloaded from https://rufus.ie/ .
:: Note, there is a portable version which does not require an installation.
set LOCRF=D:\Portable\Rufus
:: location/path of removedrive.exe
:: removedrive.exe can be downloaded from https://www.uwe-sieber.de/drivetools_e.html.  No
installation required
set LOCRD=D:\Utility
:: location/path of partitionwizard.exe
:: MiniTool Portable Partition Wizard can be downloaded from
::   https://www.partitionwizard.com/partitionmagic/portable-partition-magic.html.
::   No installation required
set LOCPW=D:\Portable\MiniToolPartitionWizard
:: location/path of Raspbian OS Image
set LOCRB=H:\Electronica\Raspberry Pi\OS\Raspbian
:: location/path of extra install shell scripts
set LOCSH=H:\Electronica\Raspberry Pi\0. Starter
:: type of Raspbian Image. My personal naming of Raspbian images goes like this
::      Raspbian_2020-02-05_Buster-Lite.img
set TYPE=lite
:: set TYPE=base
:: set TYPE=full

echo Please insert USB-stick with micro SD card into USB-slot.
echo.
pause

:: Find first removable drive
for /F "skip=1 tokens=1-10" %%A in ('wmic logicaldisk get description^, deviceid') do (
    if "%%A %%B"=="Removable Disk" (
        set FirstRemovableDisk=%%C
        goto :found
    )
)

:found
echo.
:: USB drive found?
if "%FirstRemovableDisk%" neq "" (
    rem check if FAT32
    for /f "tokens=5" %%a in ('@fsutil fsinfo volumeinfo %FirstRemovableDisk% ^| findstr /B "File
System Name : "') do (@set DriveType=%%a)
    if "!DriveType!" equ "FAT32" (
        rem get the latest image
        for /f "tokens=*" %%a in ('dir "%LOCRB%\*-%TYPE%.img" /b /od') do set IMG=%%a
        choice /M "!IMG! will be written to %FirstRemovableDisk% OK?"
        if !ERRORLEVEL! equ 1 (
            echo Writing image !IMG! onto disk %FirstRemovableDisk% with Rufus
            pushd %LOCRF%
            rufus.exe -f FAT32 -i "%LOCRB%\!IMG!"
            popd
            echo.
            echo Copying additional files to %FirstRemovableDisk%, please wait ...
            copy /Y secret              %FirstRemovableDisk%\  1>nul 2>nul
            copy /Y ssh                 %FirstRemovableDisk%\  1>nul 2>nul
            copy /Y wpa_supplicant.conf %FirstRemovableDisk%\  1>nul 2>nul
            mkdir %FirstRemovableDisk%\SetupScripts            1>nul 2>nul
            copy /Y "%LOCSH%\*.sh"      %FirstRemovableDisk%\SetupScripts  1>nul 2>nul
            ren  %FirstRemovableDisk%\SetupScripts\BaseConfigME-0.sh RunNow.sh
            %LOCRD%\removedrive E:  1>nul 2>nul
            echo.
            echo Image ready. You can now safely remove USB-stick with micro SD card!
        )
    ) else (
        echo Micro SD card not FAT32
        echo   1. Delete all partitions on Micro SD card
        echo   2. Create FAT32 formatted partition
        echo.
        %LOCPW%\partitionwizard.exe
        echo.
        echo Restart this CMD script.
        pause
```

```
    )
) else (
    echo No USB-stick or SD-card found!
)
echo.
pause
```

A Security Update For Raspbian

What has changed?
First, from now on SSH will be disabled by default on our images. SSH (Secure SHell) is a networking protocol which allows you to remotely log into a Linux computer and control it from a remote command line. As mentioned above, many Pi owners use it to install a Pi headless (without screen or keyboard) and control it from another PC.
In the past, SSH was enabled by default, so people using their Pi headless could easily update their SD card to a new image. Switching SSH on or off has always required the use of raspi-config or the Raspberry Pi Configuration application, but to access those, you need a screen and keyboard connected to the Pi itself, which is not the case in headless applications. So we've provided a simple mechanism for enabling SSH before an image is booted.
The boot partition on a Pi should be accessible from any machine with an SD card reader, on Windows, Mac, or Linux. If you want to enable SSH, all you need to do is to put a file called ssh in the /boot directory. The contents of the file don't matter, it can contain any text you like, or even nothing at all. When the Pi boots, it looks for this file; if it finds it, it enables SSH and then deletes the file. SSH can still be turned on or off from the Raspberry Pi

The risk with an open SSH port is that someone can access it and log in. To do this, they need a user account and a password. Out of the box, all Raspbian installs have the default user account 'pi' with the password 'raspberry'. If you're enabling SSH, you should really change the password for the 'pi' user to prevent a hacker using the defaults. To encourage this, we've added warnings to the boot process. If SSH is enabled, and the password for the 'pi' user is still 'raspberry', you'll see a warning message whenever you boot the Pi, whether to the desktop or the command line. We're not enforcing password changes, but you'll be warned whenever you boot if your Pi is potentially at risk.

How do I get the updates?
To update your existing Jessie image with all the bug fixes and these new security changes, type the following at the command line:

```
# will update the repository package list
sudo apt update

# will upgrade the current installed Raspbian OS to the latest version
sudo apt -y full-upgrade

# will remove old packages with changed dependencies
sudo apt -y autoremove

# will remove unneeded files
sudo apt autoclean
```

and then reboot.

Please note that installing this update on an existing Raspbian install will not change the status of SSH on that machine; if SSH is enabled, installing the update leaves SSH enabled, and vice-versa.

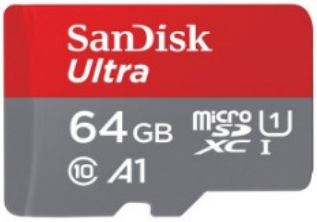
Raspberry Pi SD Card Speed Test

Since we first launched Raspberry Pi, an SD card (or microSD card) has always been a vital component. Without an SD card to store the operating system, Raspberry Pi is pretty useless*! Over the ensuing eight years, SD cards have become the default removable storage technology, used in cameras, smartphones, games consoles and all sorts of other devices. Prices have plummeted to the point where smaller size cards are practically given away for free, and at the same time storage capacity has increased to the point where you can store a terabyte on your thumbnail.

SD card speed ratings, and why they matter
However, the fact that SD cards are now so commonplace sometimes conceals the fact that not all SD cards are created equal. SD cards have a speed rating – how fast you can read or write data to the card – and as card sizes have increased, so have speed ratings. If you want to store 4K video from your digital camera, it is important not just that the card is big enough to hold it, but also that you can write it to the card fast enough to keep up with the huge amount of data coming out of the camera.
The speed of an SD card will also directly affect how fast your Raspberry Pi runs, in just the same way as the speed of a hard drive affects how fast a conventional desktop computer runs. The faster you can read data from the card, the faster your Raspberry Pi will boot, and the faster programs will load. Equally, write speed will also affect how well any programs which save large quantities of data run – so it's important to use a good-quality card.

What speed can I expect from my SD card?
The speed rating of an SD card should be printed either on the card itself or on the packaging. The 32GB card shown below is Class 4, denoted by the 4 inside the letter C – this indicates that it can write at 4MB/s.

The 64GB card shown below is Class 10, and so can write at 10MB/s. It also shows the logo of UHS ("ultra high speed") Class 1, the 1 inside the letter U, which corresponds to the same speed.



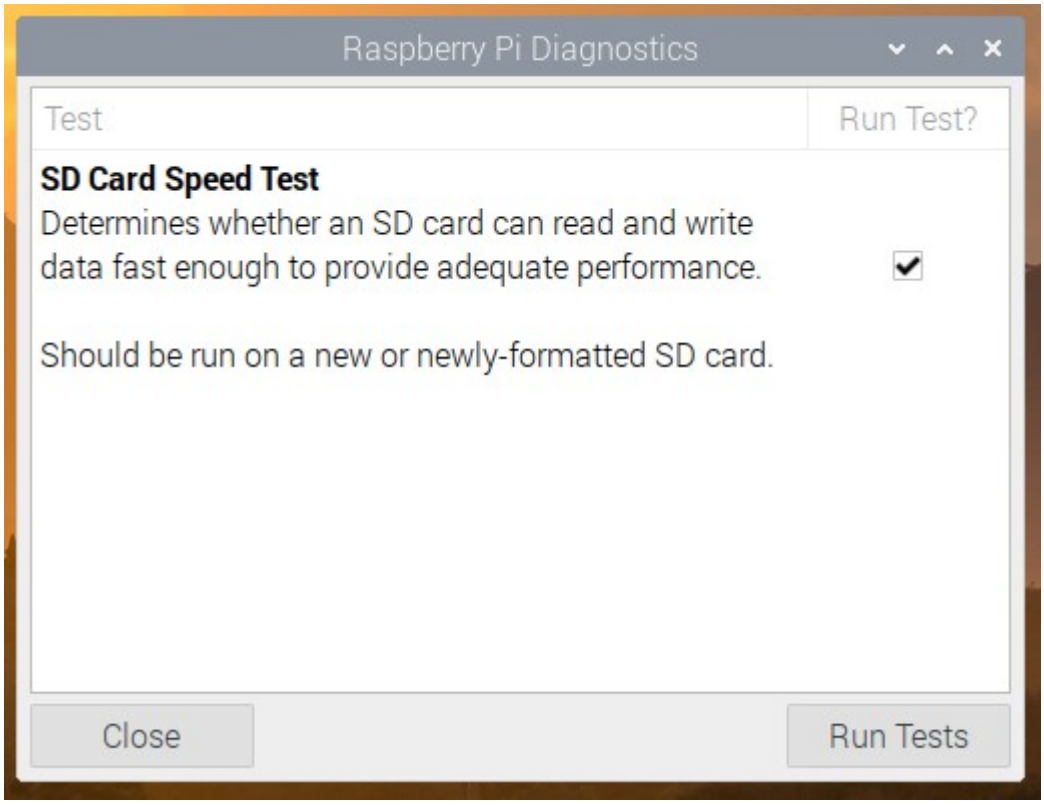
More recently, speeds have started to be quoted in terms of the intended use of the card, with Class V10 denoting a card intended for video at 10MB/s, for example. But the most recent speed categorisation – and the one most relevant to use in a Raspberry Pi – is the new A (for

"application") speed class. We recommend the use of Class A1 cards (as the one above – see the A1 logo to the right of the Class 10 symbol) in Raspberry Pi – in addition to a write speed of 10MB/s, these support at least 1500 read operations and 500 write operations per second. All the official Raspberry Pi microSD cards we sell meet this specification.

A new tool for testing your SD card speed
We've all heard the stories of people who have bought a large capacity SD card at a too-good-to-be-true price from a dodgy eBay seller, and found that their card labelled as 64GB can only actually hold 2GB of data. But that is at least fairly easy to spot – it's much harder to work out whether your supposedly fast SD card is actually meeting its specified speed, and unscrupulous manufacturers and sellers often mislabel low quality cards as having unachievable speeds.
Today, as the first part of a new suite of tests which will enable you to perform various diagnostics on your Raspberry Pi hardware, we are releasing a tool which allows you to test your SD card to check that it performs as it should.
To install the new tool, from a terminal do

```
sudo apt update
sudo apt install agnostics
```

Once installed, you will find the new application "Raspberry Pi Diagnostics" in the main menu under "Accessories", and if you launch it, you'll see a screen like this:

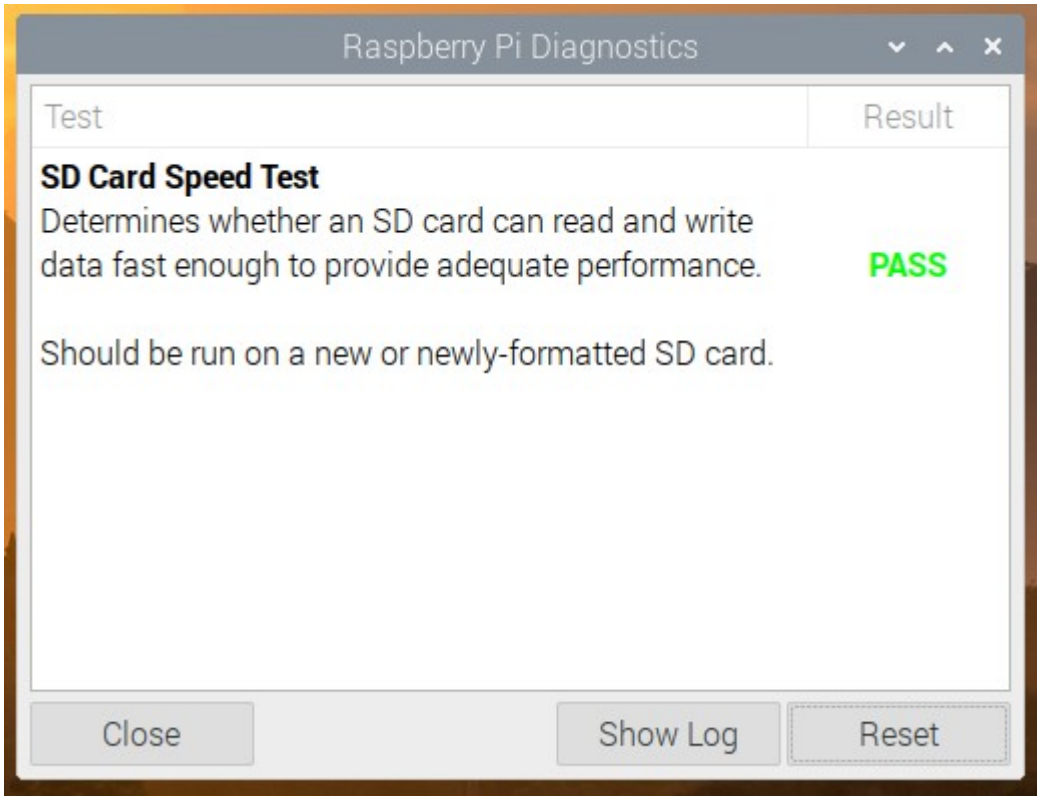


In future, this screen will show a list of the diagnostic tests, and you will be able to select which you want to run using the checkboxes in the right-hand column. But for now, the only test available is SD Card Speed Test; just press "Run Tests" to start it.

Understanding your speed test results

One thing to note is that the write performance of SD cards declines over time. A new card is blank and data can be written to what is effectively "empty" memory, which is fast; but as a

card fills up, memory needs to be erased before it can be overwritten, and so writes will become slower the more a card is used. The pass / fail criteria in this test assume a new (or at least freshly formatted) card; don't be alarmed if the write speed test fails when run on the SD card you've been using for six months! If you do notice your Raspberry Pi slowing down over time, it may be worth backing up your SD card using the SD Card Copier tool and reformatting it.
The test takes a minute or so to run on a Raspberry Pi 4 (it'll take longer on older models), and at the end you'll see a results screen with either (hopefully) PASS or (if you are less fortunate) FAIL. To see the detailed results of the speed test, press "Show Log", which will open the test log file in a text editor. (The log file is also written to your home directory as **rpdiags.txt**.)



We are testing against the A1 specification, which requires a sequential write speed of 10MB/s, 500 random write operations per second, and 1500 random read operations per second; we run the test up to three times. (Tests of this nature are liable to errors due to other background operations accessing the SD card while the test is running, which can affect the result – by running the test multiple times we try to reduce the likelihood of a single bad run resulting in a fail.)
If the test result was a pass, great! Your SD card is good enough to provide optimum performance in your Raspberry Pi. If it failed, have a look in the log file – you'll see something like:

```
Raspberry Pi Diagnostics - version 0.1
Mon Feb 24 09:44:16 2020

Test : SD Card Speed Test
Run 1
prepare-file;0;0;12161;23
seq-write;0;0;4151;8
rand-4k-write;0;0;3046;761
rand-4k-read;9242;2310;0;0
Sequential write speed 4151 kb/s (target 10000) - FAIL
```

```
Note that sequential write speed declines over time as a card is used - your
card may require reformatting
Random write speed 761 IOPS (target 500) - PASS
Random read speed 2310 IOPS (target 1500) - PASS
Run 2
prepare-file;0;0;8526;16
```

You can see just how your card compares to the stated targets; if it is pretty close to them, then your card is only just below specification and is probably fine to use. But if you are seeing significantly lower scores than the targets, you might want to consider getting another card.