How to Write Protect an SD Card

7 April 2019, http://DementiaMusicPlayer.org, Ross Porter rosswesleyporter@gmail.com

About SD card write protection TMP_WRITE_PROTECT

This step is optional. It will increase the durability of the system. If you or your friend already work with Raspberry Pis, you may find this fun. If you don't then you might find this frustrating and it's perfectly OK to skip this optional step.

About TMP_WRITE_PROTECT:

- Raspbian and DietPi, don't generally like to be rudely shut down i.e. having the power plug pulled.
- Rude power downs can interrupt writes to the micro SD card and potentially corrupt the OS.
- But we have a special case here, because DQMusicBox does not need to be online and does not need to be updated. In other words, it is safe to write protect the micro SD card.
- The SD card standard has a rarely used feature known as TMP_WRITE_PROTECT (which is a useful google search term) i.e. temporary write protection.
- When in this mode, the SD accepts write accepts write requests, but any such writes are made to temporary storage and are deliberately lost upon reboot or power loss.
- In other words, if you pull the plug on a DQMusixBox you are effectively doing a factory reset. Music stored on the USB thumb drive is unaffected.
- To enable this, set the TMP_WRITE_PROTECT bit in the Card Specific Data on micro-SD card (more on this below).
- Here is the SD Association's official description of TMP_WRITE_PROTECT from the SD Specifications Part 1 Physical Layer Simplified Specification Version 6.00: *Temporarily protects the entire card content from being overwritten or erased (all write and erase commands for this card are temporarily disabled). This bit can be set and reset. The default value is 0, i.e. not write protected.*
- For the curious, here are some of the other fields in the Card Specific Data, chart taken from the same document linked just above:

File format group	FILE_FORMAT_GRP	1	xb	R/W(1)	[15:15]
copy flag	COPY	1	xb	R/W(1)	[14:14]
permanent write protection	PERM_WRITE_PROTECT	1	xb	R/W(1)	[13:13]
temporary write protection	TMP_WRITE_PROTECT	1	xb	R/W	[12:12]
File format	FILE_FORMAT	2	xxb	R/W(1)	[11:10]
reserved		2	00b	R/W	[9:8]
CRC	CRC	7	xxxxxxxb	R/W	[7:1]
not used, always'1'	-	1	1b	-	[0:0]

Table 5-4 : The CSD Register Fields (CSD Version 1.0)

- References:
  - SD Card Write Protection
  - Build the SD Locker and Make Your SD Cards More Secure

Option 1: Build the SD locker project

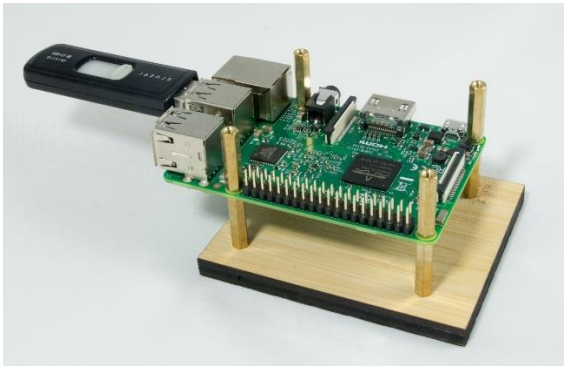
There are several ways to implement TMP_WRITE_PROTECT. I was originally inspired by the SD locker project on Hackaday and by its successor. Both of which look like fun. But I ended up using Option 2 below as it is more consistent with Raspberry Pi work.

Option 2: Use a Raspberry Pi 3

This is the option that I used. In brief, the write protection process is:

1. Write the DQMusicBox image to a micro-SD card, as per the instructions above.
2. Boot a Raspberry Pi 3 from a USB thumb drive.
3. Once booted, insert a micro-SD card and issue the following command:
   `sudo ./sdtool/static/arm-sdtool /dev/mmcblk0 lock`

This is my setup for doing so:



The slightly tricky bit is to get the Raspberry Pi 3 to boot from USB. See below for the process that I used to create to create the setup above. You will only have to do this once.

1. You'll need:
   a. **A Raspberry Pi 3**. I keep a Pi 3 just for this purpose i.e. I don't use this Pi 3 inside a DQMusicBox.
   b. **A good USB thumb drive.** I keep a thumb drive just for this purpose. Apparently this won't work with some low performance USB thumb drives, although it worked with my old 2GB thumb drive (pictured above). I now use a SanDisk Extreme 32GB thumb drive.
   c. **A micro-SD card.** You will only need for 30 minutes to complete the steps below, then you can use the card for other purposes.

2. Download DietPi.

3. Write the DietPi image to the micro-SD card (using Etcher or Win32diskimager).

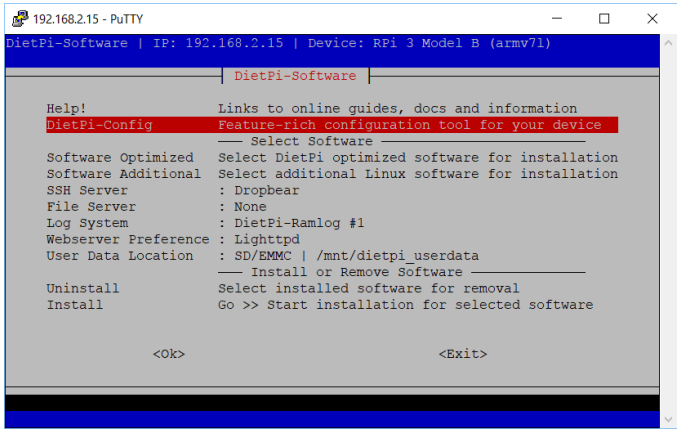
4. Write the same DietPi image to the USB thumb drive.

5. Attach Ethernet to your Raspberry Pi 3.

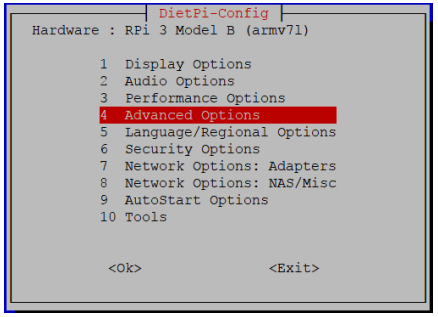
6. Insert the micro-SD card.

7. Start the DietPi installation process i.e.:
    a. Boot the Pi 3 from the micro-SD card
    b. Connect to the Pi 3 over the network with Putty or something similar, login with username=root password=dietpi

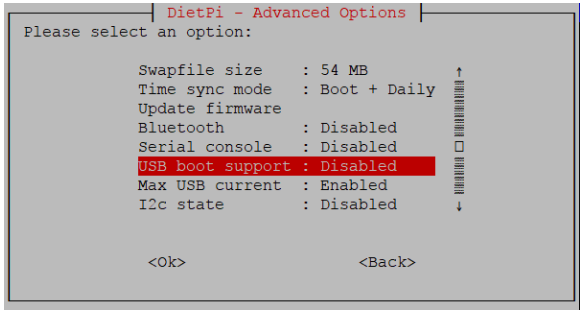
8. After a while, you will get to this screen, choose DietPi-Config



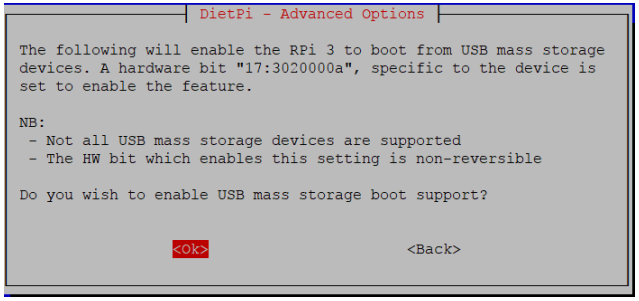
9. Choose Advanced Options:



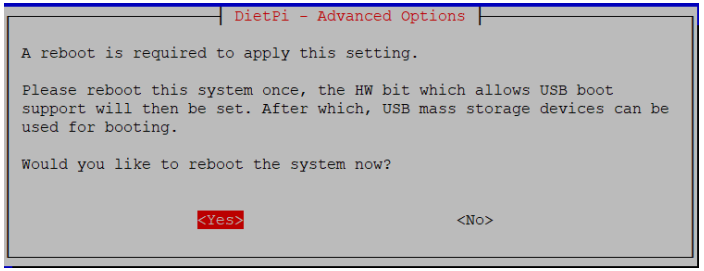
10. Choose USB boot support and choose to **enable** it:



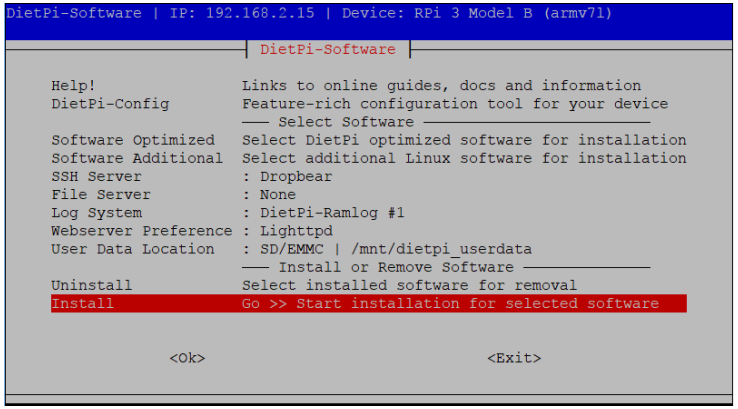
11. Choose OK (read the full message first):



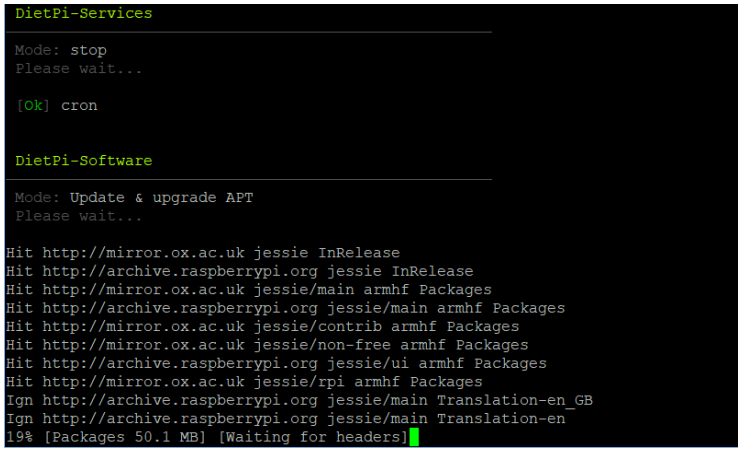
12. Choose Yes to reboot:



13. After rebooting, choose Install:

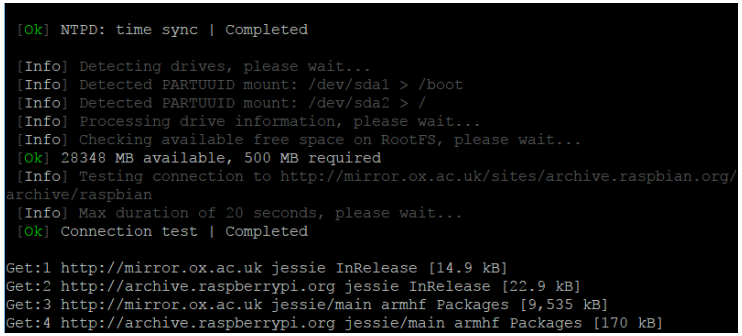


14. Let the installation run and finish.

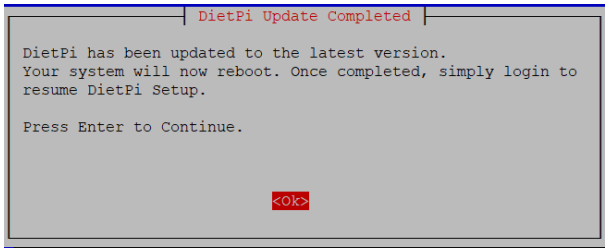


15. Once the installation is complete, switch to a USB setup:
    a. Unplug the Pi.
    b. Remove the micro-SD Card.
    c. Insert the USB thumb drive.
    d. Plug the Pi back in and boot.

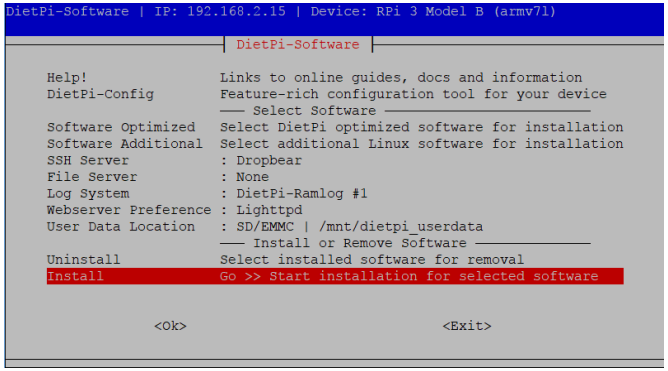
16. Install DietPi on the USB thumb drive. Start by letting DietPi update itself:



17. When prompted, restart your Pi:

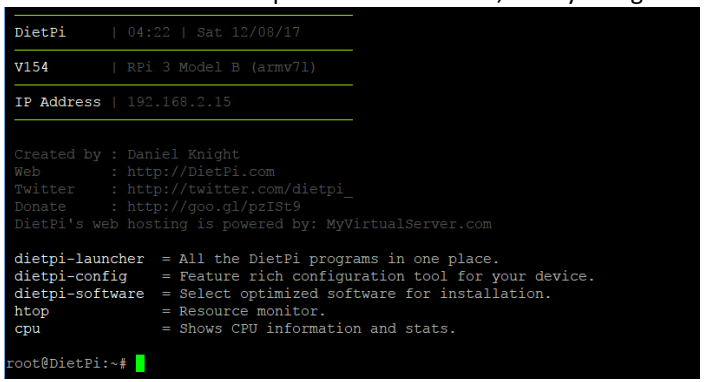


18. Install:

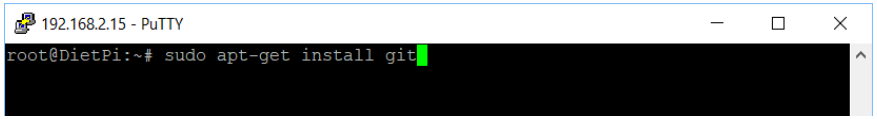


19. Let DietPi do more updating:

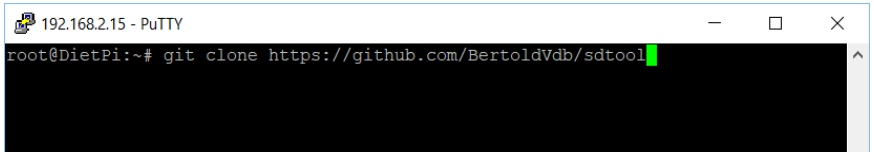
20. DietPi will reboot to complete the installation, then you'll get a normal command prompt:



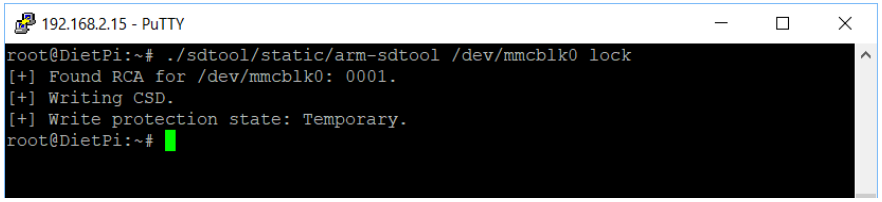
21. Install git with
    `sudo apt-get install git`



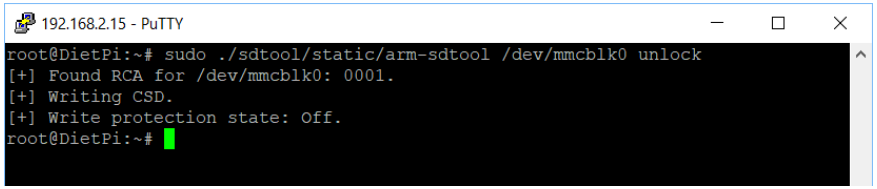
22. Get the write protection tool sdtool with
    `git clone https://github.com/BertoldVdb/sdtool`



23. Put an micro-SD card in the slot, practice enabling TMP_WRITE_PROTECT:
    `sudo ./sdtool/static/arm-sdtool /dev/mmcblk0 lock`



24. Now practice reversing the above:
    `sudo ./sdtool/static/arm-sdtool /dev/mmcblk0 unlock`



25. You are done!