# Raspberry Pi: Prepare SD Card

***Note**: The term SD card mentioned here generally covers/means the microSD card.*

This is a guide to prepare such card from scratch. If you already have the one that comes along with the board (NOOBS card, is it?), this procedure will override and delete existing content. Backup or forever hold your peace! Well, if you cannot do that and you want to have NOOBS back, head on to this page and download the latest NOOBS image.

## Bare-metal Programming

***Note:** Your instructor will show you how to partition/format the SD card.*

To exclusively work on bare-metal programming (no OS):

- Format the card with at least one FAT32 partition
  - Linux platform - use *fdisk* for this (usually need root... unless your user have permission)
  - Windows platform - use *diskpart* (run as administrator)... OR, maybe *this*? (The link is for portable version.. there is also an installer if that is your cup of coffee 😎)
- Get the required firmware from here - we only need 2 files from boot folder (bootcode.bin and start.elf). For the sake of being 'politically correct', I also copied LICENSE.broadcom file. OR, get my personal copy here (which is what I usually pass to my students on a USB drive).
- Place those 3 files in the root of the previously FAT32-formatted SD card
- Along with your kernel.img code, you should be able to take control of the Pi

A nice config for multiple kernels - this is what I use to have multiple kernel (my1barepi codes) images on my SD card

config-my1barepi.txt

```
# kernel is the alternative kernel filename
# - [Pi 1, Pi Zero, and Compute Module] kernel.img
# - [Pi 2, Pi 3, and Compute Module 3] kernel7.img
# - [Pi4] kernel7l.img.
#kernel=kernel_video_temp.img
#kernel=kernel_sdcard.img
kernel=kernel_pick_one.img
```

## Installing Raspbian OS

***Update20210907** The official OS is now known as Raspberry Pi OS (instead of Raspbian). Check out this official page. I will try to update this site A.S.A.P. - but, do not hold your breath* 😬

We will be using Raspbian (the official Linux distribution for Raspberry Pi). This enables us to run web servers and other network-related stuffs.

***[201804011654]*** *Note: I just noticed there is now an option to use Windows10 IOT Core (which is prepared by Microsoft as a third party option), but I will not be using that here. Checkout this page for other options*

- Download system image
  - Official Raspbian (latest)
    - This is a ZIP file containing an image file - e.g. 2018-11-13-raspbian-stretch.zip
  - For PGT302 students, get one with PGT302-specific customization
    - Ask your instructor
- Extract (unzip) the image (file with *.img extension)
  - e.g. 2018-03-13-raspbian-stretch.img
  - currently, at least 8GB SD card is required...
- Write the image to SD card
  - Insert the SD card to your SD card reader
  - Linux Platform - use *dd* for this
    - assuming the device is at /dev/sdb
    - use `dd if=2018-03-13-raspbian-stretch.img of=/dev/sdb` and wait...
  - Windows Platform - I recommend Win32DiskImager (I got the Win32DiskImager-1.0.0-binary.zip file)
    - find (and VERIFY) drive letter for your SD card
    - select image, start write and wait...

# Raspbian-ready Plus Bare-metal Programming

To have Raspbian-ready card as well as working on bare-metal programming:

- Follow normal instruction for installing Raspbian OS

***Note:*** *Do the following procedure on your PC.*

To use bare-metal code:

- Rename kernel.img and config.txt to avoid being used
  - e.g. rename kernel.img → kernel-raspbian.img
  - e.g. rename config.txt → config-raspbian.txt
- Make sure there is no config.tx file
- Simply copy your own compiled kernel.img to the FAT32 partition

To get Raspbian back running:

- Get the saved files appropriately named:
  - e.g. copy kernel-raspbian.img → kernel.img
  - e.g. copy config-raspbian.txt → config.txt

# Raspberry Pi Zero as USB Client

Pi Zero has a USB On-the-Go (OTG) hub - which, basically means that it can be both host (like USB hub on a PC) AND client (like USB hub on Android or most gadgets these days). So, to setup Pi Zero as a client (this is done on a PC - while preparing the card),

1. Follow normal instruction for installing Raspbian OS
2. Edit `config.txt` (on boot partition) and insert `dtoverlay=dwc2` line
3. Edit `cmdline.txt` and insert `modules-load=dwc2,g_ether` kernel parameter
    - also, insert `g_ether.host_addr=<mac_addr>` to get a fixed MAC address (easier to manage!)
4. Add empty file called `ssh` (same location as `config.txt`) - this will enable ssh

Connect Pi Zero to a PC (allow some time for it to finish booting) and it should appear as ethernet device. To connect to it,

- using network manager:
    - config ipv4 as link-local only
    - config ipv6 as ignore
- use ssh to connect
    - ssh pi@raspberrypi.local
        - default password: raspberry
    - use ssh-copy-id to use key-based auth

To share the internet with the Pi

- find the IP address on usb0

```
$ ifconfig usb0
```

    - we need this later on Pi
- allow IP forwarding on the host

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

    - to remove this later

```
echo 0 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
```

    - *note for wifi internet, change `eth0` to `wlan0`*
- ssh into pi
- (OPTIONAL) add a name server (e.g. nameserver <ip-add-of-usb0>) to /etc/resolv.conf
    - or, just use 8.8.8.8
- set the default gateway on Pi to host PC's IP

```
# route add default gw <ip-add-of-usb0>
```

Have fun!

***Note:****I have tested this using Raspbian Lite on Pi Zero and it worked! (Obviously)*

# Advanced Configurations

Using the above setup should get simple projects going without any problems. However, some things on the BCM2835 require special configurations on the GPU side. This can be changed by having a `config.txt` file in the same path as the files above.

Example of the configuration file:

config.txt

```
# --------------
# MEMORY OPTIONS
# --------------

# specify gpu memory allocation
# - min 16, max 192 (256), 448(512), 944 (1024)
# - default 64
#gpu_mem=64

# disables CPU access to GPU L2 cache
# - default 0 (enabled)
#disable_l2cache=1

# --------------
# KERNEL OPTIONS
# --------------

# specify kernel name
# - kernel7.img default for pi2/pi3
# - kernel8.img preferred on pi3 (for 64b mode)
# - common default is kernel.img?
#kernel=kernel.img

# specify startup address for ARM kernel
# - default 32b: 0x8000
# - default 64b: 0x80000
# - kernel_old=1 option overrides to 0?
#kernel_address=0x8000
#kernel_old=1

# ----------------
# ADVANCED OPTIONS
# ----------------

# camera needs start_x.elf firmware
# - or, start_file=start_x.elf, fixup_file=fixup_x.dat
#start_x=1
```

```
# prevent red camera led to turn on while camera is active
# - default 0 (enabled)
#disable_camera_led=1
```

To activate an option, simply remove the '#' character for the beginning of the option line (uncomment). More information on `config.txt` can be found here. We can also have conditional filters.

Other custom configuration(s):

config.txt

```
# for 5-inch lcd with touchscreen
max_usb_current=1
hdmi_group=2
hdmi_mode=87
hdmi_cvt 800 480 60 6 0 0 0
dtoverlay=ads7846,cs=1,penirq=25,penirq_pull=2,speed=50000,keep_vref_on
=0,swapxy=0,pmax=255,xohms=150,xmin=200,xmax=3900,ymin=200,ymax=3900
#display_rotate=0
```

# General Issues

### MicroSD Card Failed Boot

Sometimes, the card simply cannot boot. Use fdisk to check/create 255 heads, 63 sectors and calculate the required cylinders based on

```
disk_size = cylinders * head * sector * sector_size
```

Not sure why this happens... maybe BIOS issue when formatting on PC? Or, maybe my students' laptops are infected with virus?

### Raspbian Update Error

- I used the 2017-09-07-raspbian-stretch.img and got an error while trying to update
  - problems seem to be with storage space (98% usage)
  - turns out the partition for root fs was only 4.9GB
- to resize the partition, use fidsk
  - assume card is /dev/sdb (i use USB card reader)
  - `fdisk /dev/sdb`
  - delete partition 2 and recreate (make sure use the same start sector!)
  - save and exit
  - clean the fs `e2fsck -f /dev/sdb2`
  - resize `resize2fs /dev/sdb2`

From:
https://azman.my1matrix.cc/ - **Azman @MY1**

Permanent link:
**https://azman.my1matrix.cc/doku.php?id=raspi:raspi_prepare_card**

Last update: **2026/02/08 03:46**