Raspberry Pi: Prepare Raspbian

***Note**: This assumes the previous how-to on preparing the SD card has been covered and Raspbian is already 'installed' on the SD card.*

Booting Raspbian

- Simply insert the SD card and power up
  - It will take some time, but a Linux Desktop will appear on screen.
- Initial Configuration:
  - locale set to US
  - timezone changed to Asia/Kuala_Lumpur
  - keyboard layout set to US (English)
- Run update `sudo apt update` and upgrade `sudo apt upgrade` if necessary
  - if error occurs, most probably problem with the mirror (change to another mirror)

Getting the tools

- Get kernel headers (compiling loadable kernel module)
  - *`sudo apt install`* `raspberrypi-kernel-headers`
- To be able to compile codes that require sqlite
  - *`sudo apt install`* `sqlite3 libsqlite3-dev`
- To have web server with php
  - *`sudo apt install`* `apache2 php php-cgi libapache2-mod-php php-sqlite3`
  - edit `php.ini` to enable pdo support
  - enable mod_rewrite (create link in mods-enabled from mods_available)
    - look for `rewrite`
  - default path for web is `/var/www/html`
- I also want `screen` and `ntfs-3g` (ntfs with r/w)
  - *`sudo apt install`* `screen ntfs-3g`

Prepare for GUI Development

This is about doing GUI development while running Raspbian on Pi itself.

- install glade (will also get gtk library)
  - *`sudo apt install glade`*

Using Camera & Image Processing

- to access camera, load the Video4Linux kernel module
  - *`sudo modprobe bcm2835-v4l2`*
  - not needed if you enabled camera using Configuration Page

- install ffmpeg stuffs
  - *sudo apt install libavcodec-dev libavdevice-dev*
- if going for GTK
  - *sudo apt install libgtk2.0-dev*
- if going for SDL
  - *sudo apt install libsdl1.2-dev*

Get some codes

- Get my source codes (e.g. download everything @/home/pi/Work)

```
mkdir -pv /home/pi/Work && cd /home/pi/Work
git clone git://github.com/azman/my1codelib.git
git clone git://github.com/azman/my1webapp.git
git clone git://github.com/azman/my1apisrv.git
git clone git://github.com/azman/my1linuxpi.git
git clone git://github.com/azman/my1termu.git
git clone git://github.com/my1matrix/my1goose.git
```

- Obviously, we need `git` for this
  - *sudo apt install* `git`

Upgrading Installed Packages

To upgrade locally installed sofware:

- Update apt database

```
$ sudo apt update
```

- Upgrade local packages

```
$ sudo apt upgrade
```

If packages got held/kept back, simply reinstall those packages:

- Reinstall

```
$ sudo apt install --reinstall
```

Upgrading Raspbian Version

To do an upgrade (e.g. I did this to upgrade stretch to buster):

- Modify `/etc/apt/sources.list` and replace the release codenames (e.g. change stretch to

buster)
    - do the same for `/etc/apt/sources.list.d/*.list`
- Update package list

```
$ sudo apt update
```

- Upgrade distribution

```
$ sudo apt dist-upgrade
```

- Do house cleaning

```
$ sudo apt autoremove
$ sudo apt clean
```

That should do it!

Making Pi Visible on Local Network

Note: *This is already in the default install.*

To make Pi hostname visible on local network, get avahi-daemon (default install)

- *`sudo apt install`* `avahi-daemon`
- this is part of Bonjour/Rendezvous/ZeroConf multicast DNS (aka mDNS) services
- change hostname as desired
    - use raspi-configuration tool
- clients need to support mDNS as well:
    - windows need this
    - linux require avahi (for slackers, slackbuilds.org got this)