Lab Work 4 - Using Embedded OS

This module covers working with Raspbian Linux on the Raspberry Pi. We will see how we can access the hardware resources from within Raspbian. We will also see how we can utilize a web server running on Pi.

Preparing 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)

2026/02/09 22:23

Running Codes in my1linuxpi

coming soon...

Archive

Some old notes.

Starting NOOBS: Installing Raspbian

***Note**: Since this took too much time to be demonstrated during a lab session, NOOBS will no longer be considered/discussed. However, it is still available on the official download page. This is left here for reference.*

Installation of the OS actually involves in creating more partitions for boot, root and optionally data - the original single partition is reserved/maintained as 'recovery' partition. This is my fdisk output for the NOOBS card AFTER finishing the installation.

```
Disk /dev/sdc: 7892 MB, 7892631552 bytes
4 heads, 16 sectors/track, 240864 cylinders, total 15415296 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000031bb

   Device Boot      Start         End      Blocks   Id  System
/dev/sdc1            8192     3011718     1501763+   e  W95 FAT16 (LBA)
/dev/sdc2         3014656    15349759     6167552   85  Linux extended
/dev/sdc3        15349760    15415295       32768   83  Linux
/dev/sdc5         3022848     3145727       61440    c  W95 FAT32 (LBA)
/dev/sdc6         3153920    14292991     5569536   83  Linux
/dev/sdc7        14295040    15343615      524288   83  Linux
```

The recovery partition is at /dev/sdc1 (notice that it is using FAT16 partition). The boot partition is in /dev/sdc5, which is using FAT32. The root partition is at /dev/sdc6, while a data partition is created on /dev/sdc7. Note that /dev/sdc{5,6,7} are all logical partitions made from primary partition at /dev/sdc2 (look at the sector address Start and End). Finally, partition /dev/sdc3 is where NOOBS keeps its configuration files.

Your instructor will give a demonstration on how NOOBS is used to configure an OS for Pi.

Proceeding with (ONLY) Raspbian

***Note**: Raspbian has grown considerable since this was written, and no longer suitable for my lab session.*

Using NOOBS is nice, but somehow Pi simply will not boot without a keyboard attached. Installing Raspbian image (and only Raspbian) on the (micro)SD card is supposed to remedy that problem. It is a zipped image file which is supposed to be written directly to the card (using dd on Linux). You can get a manual on how to work with this image file from here.

If you have problems downloading 900+MB zip file (like me... it takes forever!), we can actually extract the relevant files from the NOOBS card. Look into the recovery partition, and find two files from folder 'raspbian' - boot.tar.xz and root.tar.xz - which are the only files we need. I created two partitions:

- boot partition (FAT32) - extract boot.tar.xz (MD5) into this partition
- root partition (ext4) - extract (duh!) root.tar.xz (MD5) into this one

Note that those files available locally on our server were extracted from the older NOOBS 1.3.9. Newer versions are usually larger in size and I want to keep it small - this version is the one that came with the SD Card purchased along with the board.

When you first boot up to Raspbian, it will run raspbian-config (allows you to configure a few things before starting up). As mentioned earlier, we can set Raspbian to boot to Scratch here, but the other thing that I would recommend to have is to start the SSH server running. This will allow us to remotely login through SSH (a network is obviously required) in case we do not have a suitable display 😎

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

Permanent link:
https://azman.my1matrix.cc/doku.php?id=archive:pgt302lab03

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