

# Smart-Building Report

Simon Remington

January 26, 2019

Contents

1 Introduction.....2

|  |   |
|--|---|
| 2 Setting-up Raspberry Pi.....                       | 2 |
| 2.1 Obtaining the Raspberry Pi Operating System..... | 2 |
| 2.2 Windows flashing .....                           | 2 |
| 2.3 Linux flashing.....                              | 4 |
| 2.4 The Secure Shell.....                            | 4 |
| 3 Docker .....                                       | 5 |
| 4 this is section .....                              | 6 |
| 4.1 this is sub .....                                | 6 |

## 1 Introduction

This is the report to for Smart Building Project. It covers the procedures I have taken so far.

This report is not a tutorial, nor does it cover all the rabbit holes and dead end to achieve my goal.

## 2 Setting-up Raspberry Pi

### 2.1 Obtaining the Raspberry Pi Operating System

The RPI image can be obtained from:

<https://www.raspberrypi.org/downloads/raspbian/>

The image is: Raspbian Stretch Lite, a minimal image based on Debian Stretch. The lite image is for a headless install. All communication to the RPI is made via ssh.

Version: November 2018

Release date: 2018-11-13

Kernel version: 4.14

Download and unpack the image.

I have used both Windows and Linux Debian to create the bootable operating system so that I get a greater knowledge and understanding of completing the flash process.

### 2.2 Windows flashing

Check the hash, SHA-256 of the image.

Use Windows built-in certUtil -hashfile Path/To/File/file.img SHA256 to compute the hash. Obtainable on both Windows 7 and 10 machines.

Insert the SD card into a Window's machine.

Use Etcher to flash image, obtained from:

[https : //www.balena.io/etcher/](https://www.balena.io/etcher/) Download

Etcher and install.

Using Etcher, select source (Path to .img), Destination (Tath to SD card) Flash the card.

Dismount then remount SD card. Donot repair the card when asked to by Windows.

CMD into the card then create an empty file named ssh with no extension, this can be accieved by entering the following command at the prompt:

```
C:\>type nul > ssh
```

This will give secure shell access at boot time.

Info can be found at:

[https : //www.raspberrypi.org/documentation/configuration/wireless/headless.md](https://www.raspberrypi.org/documentation/configuration/wireless/headless.md)

Create a file named wpa\_suplicant.conf Info

can be found at:

[https : //manpages.debian.org/stretch/wpasupplicant/wpa\\_suplicant.conf.5.en.html/](https://manpages.debian.org/stretch/wpasupplicant/wpa_suplicant.conf.5.en.html/)

[https : //core – electronics.com.au/tutorials/raspberry – pi – zerow – headless](https://core-electronics.com.au/tutorials/raspberry-pi-zero-headless-wifi-setup.html)

[– wifi – setup.html](https://core-electronics.com.au/tutorials/raspberry-pi-zero-headless-wifi-setup.html) Add following to the file:

```
ctrl_interface=DIR=/var/run/wpa_suplicant GROUP=netdev  
update_config=1 country=IE
```

```
network={ ssid="Your-network"  
          key_mgmt=NONE # Use for no password
```

```
network={ ssid="Your-network"  
          key_psk="Your-password"  
        }
```

Now boot the Raspberry Pi, this SD card can be used in any machine that recognises the Debian Stretch OS.

This card if for the arm6 instruction set of the chip, it is capable of running in machines with the arm7 achitecture, but not the other way around.

## 2.3 Linux flashing

Download Stretch.img and unpack.

Check hash at the prompt with: user@host:~\$  
sha1sum /Path/To/file.img

Find the SD card. user@host:~\$  
df -h Dismount SD card.  
user@host:~\$ umount /dev/The  
SD card

Run the following command: user@host:~\$ sudo dd bs=4M status=progress  
if=~/Path/To/stretch-lite.img of=/dev/The SD card

Info can be found at: [https :  
//elinux.org/RPi\\_Easy\\_SD\\_Card\\_Setup](https://elinux.org/RPi_Easy_SD_Card_Setup)

When SD card is written, mount the card and create files ssh, WPA  
user@host:~\$ touch ssh  
user@host:~\$ touch wpa\_supplicant.conf

Use an editor and create file contents as above for Windows.

## 2.4 The Secure Shell

Find ipaddress of Pi, look into router settings  
Download and install Bitvise Available at: [https :  
//www.bitvise.com/](https://www.bitvise.com/)

ssh to Raspberry Pi eg ssh pi@network address change  
password from raspberry to your choice of passwd sudo  
raspi-config expand file system reboot apt update &&  
upgrade

## 3 Docker section

### 3.1 What docker is

Docker is a container system sitting on top of the operating system, it hosts an operating system of the developer's choice e.g. Raspberry PI hosting a Debian stretch OS. A Docker container sitting on top of the OS. The container hosts the guest OS perhaps a minimal Alpine image or Ubuntu image or a database, various images exist. NB The Alpine image is approximately 45MB

The Developer can build the guest OS as desired needing only to write a Docker file containing the naming of the desired OS followed by the standard commands e.g. `ap-get update && apt-get upgrade`.

Docker can be installed on the RPI with:

```
$curl -fsSL https://get.docker.com -o get-docker.sh $sudo sh get-docker.sh Info at:  
https://docs.docker.com/install/linux/docker-ce/debian/#install-using-the-convenience-script
```

Problem for Pi zero is that the latest docker is not friendly toward the arm6 with a segment error and did not work so I eventually got it going by following this advice in the following forum.

```
https://github.com/moby/moby/issues/38175
```

at prompt: `$apt-cache madison docker-ce`

We can now see the packages

```
$sudo apt-get purge docker-ce
```

```
$sudo apt-get install docker-ce=<VERSION_STRING>
```

## 4 this is section

### 4.1 this is sub