WaveOS™ Development Guide

Release 0.6.0a

Author(s): Make it Wave Ltd

Aug 26, 2019

CONTENTS

Wave® Smart Home Hotspot™ - HSPA+/LTE/5G/StarLink

CHAPTER

ONE

RELEASE NOTES AND NOTICES

This section provides information about what is new or changed, including urgent issues, Software & documentation updates, maintenance and new releases.

- 'Updates' are the term used to describe significant changes to our public source code. These technical documents are now contained within our public source code.

1.1 Version 0.6.0

- Jumped from 0.1.x to 0.6.x
- This document wasn't updated with the softwares development, instead the blog was document the manual build proceedure.
- The blog was used to show progress and create a quick point of reference for troubleshooting.
- The methodology for development is now more clear to the development team now, so the supporting documentation revision process can be adheared to.
- Moving forward this document will be easier to maitain than a blog, because there now exists a pattern in the configuration and build process.
- This document will now contain all the main configuration files and installation proceedure of the applications and interfaces.

1.1.1 Older Versions

See below a table containing earlier version of this document:

Table 1: Table 1.0 - Older Versions of this Document

archive date	version	description	download link
2019-02-11	0.1.0	also known as 0.0.1	2019-02-11_wave-software-developers_guide_v0.1.1

1.1.2 Version 0.1.0

- Incorrect usage of version control - says 0.0.1 not 0.1.0
- Firt draft only, contained no useful guides for developers, since the development and manual build process was still uncertain.

1.2 Known and Corrected Issues

Below is a table of pending issues which have been reported to our team. These issues will be cleared from this list as and when they are remedied.

Table 2: Table 1.1 - Known Issues

date	version	subject	description
24-08-2019	0.0.1	404	Reported in Google Search Console.
01-11-2018	0.0.1	N/A	no doubt many issues to report - first draft only

Comments - none

1.3 Recently Updated Topics

Nothing significant to report

CHAPTER TWO

INTRODUCTION

Welcome to the WaveOS™ developers guide. WaveOS™ is a free and open-source software. The purpose of WaveOS™ is to make Internet and IPTV access Free. This document guides you through maintaining and developing the operating system and its applications.

For more information visit https://makeitwave.com

CHAPTER

THREE

MANUAL BUILD PROCESS (ALL DEVICES)

3.1 Intro

WaveOS (Version 0.6.x - branch a) is based on the Raspbian Stretch Lite OS. This will be the base OS for all hardware variations of the Wave Smart Home Hotspot e.g. LTE, 5G etc

3.2 Step 1: Preperation

1. Download the latest version of Raspbian Stretch Lite from [Raspberrypi.org](https://www.raspberrypi.org/downloads/raspbian/).
2. Copy the latest copy of Raspbian Stretch Lite to the MicroSD Card using a program like [Etcher](https://www.balena.io/etcher) or [Win32 Disk Imager](https://sourceforge.net/projects/win32diskimager).
3. Insert the MicroSD Card into your device and connect the keyboard. Connect the device to a display e.g. via HDMI (making sure the display is turned on before applying the power cord)

# Ensure your device has a heatsink, since it will be overclocked and running at maximum capacity a lot of the time. # Ensure the power supply is the correct recommended voltage and current for the device. # Ensure the quality of the MicroSD Card (and the Power and HDMI cord) are the best quality possible.

3.3 Step 2: Wi-Fi, SSH & Update/Upgrade

1. Once the Pi has booted up, login with the default username and password :

   ```
   Username: pi
   ```

   ```
   Password: raspberry
   ```

2. Then enter setup in order to enable SSH and connect to another Wireless Hotspot e.g. your Cellular dongle Hotspot :

   ```
   sudo raspi-config
   ```

3. Via your connected display (or a remote SSH terminal client over Wi-Fi e.g. [Putty](https://www.putty.org/)) you must update and upgrade the OS :

   ```
   sudo apt-get update && sudo apt-get upgrade -y
   ```

   ```
   sudo rpi-update
   ```

4. Reboot the system :

   ```
   sudo shutdown -r now
   ```

3.4 Step 3: Cellular to Wi-Fi

This segment explains how to obtain internet access from the cellular dongle over USB and then enable a Wireless Access Point using the Raspberry Pi's onboard Wi-Fi. Before proceeding with the steps below, the device should already be connected to the internet via Ethernet or using the onboard Wi-Fi to connect to another Wireless Hotspot.

3.4.1 1st - Cellar over USB

1. Install the software with these commands :

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get install ppp usb-modeswitch wvdial
   ```

2. Poweroff the system with the following command :

   ```
   sudo poweroff
   ```

3. Once the system has powered off, Connect the cellular dongle via USB then reboot.

3.4.2 2nd - Wi-Fi Access Point

For the Wireless Access Point we will be using [RaspAP](https://github.com/billz/raspap-webgui). This application is ideal, since it features a user interface so the end-user can change the Access Point Name and Password. And the best thing is, RaspAP is now a one-line installation now.

```
wget -q https://git.io/voEUQ -O /tmp/raspap && bash /tmp/raspap
```

To see the default SSID and password, please refer to the GitHub page above. Then access the UI (http://10.3.141.1/ - username/password = admin, secret) and set the Wave SSID and password.

```
#SSID: Wave-Hotspot
```

```
#Password:  makeitwave2020!!
```

The system should now be receiving internet from the USB and broadcasting it as a Wireless Hotspot Access Point.

3.5 Step 4: Emulation Station

This segment explains how to install the Emulation Station Gaming System.

3.5.1 1st - Set the minimum amount of RAM to the GPU

```
sudo nano /boot/config.txt
```

```
# add or replace "gpu_mem = 32"
```

```
# if you skip this step, you will probably get "out of memory" errors
when compiling # Reboot to apply GPU RAM changes
```

3.5.2 2nd - Install Dependencies

```
sudo apt-get install -y libboost-system-dev libboost-filesystem-dev
libboost-date-time-dev libboost-locale-dev libfreeimage-dev
libfreetype6-dev libeigen3-dev libcurl4-openssl-dev libasound2-dev
cmake libsdl2-dev
```

3.5.3 3rd - Compile and Install

```
git clone https://github.com/Aloshi/EmulationStation

cd EmulationStation

mkdir build

cd build

# On the RPi 2, you may need to add "-DFREETYPE_INCLUDE_DIRS=/
usr/include/freetype2/".

# See issue #384 on GitHub for details.

cmake ..

# you can add -j2 here to use 2 threads for compiling in
parallel (depending on how many cores/how much memory your
RPi has)

make -j2
```

```
#If you want to install emulationstation to /usr/local/bin/
emulationstation, which will let you just type 'emulationstation' to
run it, you can do:
```

```
sudo make install
```

NOTE: This will conflict with RetroPie, which installs a bash script to /usr/bin/emulationstation. Otherwise, you can run the binary from the root of the EmulationStation folder:

```
../emulationstation
```

3.5.4 4th - Reset GPU RAM and Reboot

```
sudo nano /boot/config.txt

# change/add "gpu_mem = 32" to "gpu_mem = 128" or "gpu_mem = 256",
depending on your Pi model

sudo reboot
```

CHAPTER

FOUR

DOCUMENT AUTHOR(S):

4.1 Make it Wave Ltd