AVMultiPhone — Improve and Win

Egg, June 30, 2020: *The following blog is about what the new version of AVMultiPhone offers. It also shows how AVMultiPhone can be adapted to your own needs and there is a prize of 500 Euros each for two functions that are currently missing.*



Up to 20 hours runtime with new version

The new version of 30 June includes the deep sleep mode. This allows the AVMultiPhone to run for up to 20 hours without recharging. The sleep mode is activated by pressing the power key. This is only possible when the AVMutliPhone is running in battery mode. If the AVMultiPhone is charged via USB, the sleep mode makes no sense, as there is sufficient power available.

By pressing the power button again, the phone wakes up from deep sleep. The desktop is directly available. If you want to unlock the AVMultiPhone, you can do so by locking the screen first (lock screen in the main menu). Depending on the length of the “deep sleep” it may be necessary to unlock WLAN again (fourth icon from the left). And another thing, if you want to trigger the deep sleep via the console, you can do this via ‘pm-suspend’.

Camera “obscura”

Also included in the current release are the necessary kernel modules for the main camera. Via the menu ‘Graphics’ and ‘Camera’ a picture can be taken at any time. However, at the moment only the resolution 1280×720 pixels is possible, and there is no preview. This may not seem much, but the camera does not run on any other PhinePhone distribution.

The current camera script is published here:

```
#!/bin/bash

pfad="/home/archivista/Desktop"
dat=`date +'%Y-%m-%d-%T'`
file="$pfad/camera-$dat.jpg"
/usr/bin/media-ctl -d /dev/media1 --set-v4l2 '"ov5640
3-004c":0[fmt:UYVY8_2X8/1280x720]'
/usr/bin/ffmpeg -s 1280x720 -f video4linux2 -i /dev/video1 -
vframes 1 $file
chown archivista.archivista $file
```

The camera itself is capable of recording images with 5 megapixels, but the relevant information is currently missing. Unfortunately the parameters cannot be easily adjusted with other pixel numbers and a transfer like “ov5640 3-004c”:0[fmt:UYVY8…] is just not self-explanatory.



Adapt AVMultiPhone to your own needs

First of all, AVMultiPhone can be adapted to your own needs. The question is rather how to prevent accidentally damaging the system or creating fuses. But first, to work efficiently, we will show how AVMultiPhone can be accessed with standard Linux tools.

Access via SSH and VNC

In contrast to other PinePhone distributions, access is initially not restricted. Both SSH and VNC are available. The easiest way to access them is via the USB cable. As soon as

the AVMultiPhone is attached, the internal IP address 172.16.42.1 is set in AVMultiPhone. Ideally (i.e. under your Host) usb0 can be activated and SSH access is available. Command for this:

```
ifconfig usb0 172.16.42.2 up
ssh root@172.16.42.1
```

The ‘root’ password is 1234, and access is also possible with the ‘archivista’ user. This user can be used to start graphical programs. The only thing you have to make sure is that ‘export DISPLAY=:0’ is set in each case. Example for setting up the VNC server (archivista user):

```
export DISPLAY=:0
x11vnc -passwd archivista
```

Afterwards, the connection can be established from the host computer:

```
vncviewer -encodings 'tight copyrect' 172.16.42.1
```

Of course, other VNC programs can also be used and access via WLAN is also possible. For this reason, it is recommended to change the default passwords or to switch off the ssh service on the way. To do this, you can use the command ‘/etc/init.d/ssh stop’ or ‘rc-update del ssh default’ to permanently disable SSH.

Installing software

AVMultiPhone is based on pmOS (postmarketOS) or Alpine Linux. So all packages are available, which are also available there. These are managed with the ‘apk’ package manager. The following commands are useful:

apk info Lists all installed packages
apk list –verbose listing of all available packages
apk search xxx Search for the package xxxx
apk add –force –allow-untrusted xxx Install the pact xxx (also locally)
apk del xxx Remove the package
apk update Update package sources
apk upgrade Install newly available packages (at your own risk)

Save system

Unfortunately, the command ‘apk upgrade’ can lead to a partial or total loss of the system, because not all packages are always matched. This is especially tricky with ‘u-boot-pinephone’ and the kernel ‘linux-postmarketos-allwinner’. Both overwrite the file ‘/boot/sun50i-a64-pinephone-1.1.dtb’. This can lead to the fact that after that e.g. no sound or no SIM card is available anymore. The current remedy is to copy back the delivered ‘dtb’ file:

```
cp /boot/sun50i-a64-pinephone-1.1.old cp /boot/sun50i-a64-
```

pinephone-1.1.dtb

Afterwards, a restart must be performed, the sound and SIM card should now work again. A side note on the problem: If the situation was understood correctly, the PinePhone should have a separate ‘dtb’ file for each hardware version, but this feature is currently missing under pmOS.

Create Master Image

Uploading a new image to an SD card is easy. Apart from **etcher.io,** which does the job graphically, here again the command for the console is presented:

gunzip -c postav_200630.img.gz | dd of=/dev/sdX bs=64M

Attention: Replace /dev/sdX with the correct letter of the SD card (e.g. /dev/sdh). A wrong letter will destroy your own system!

It becomes a bit more difficult if you want to backup a current image. Copying the medium with ‘dd if=/dev/sdX of=one.img bs=64M’ works fine, but it leads to infinitely large files. A small script was created for this:

```
#!/bin/bash

umount /media/archivista/*
device="/dev/sdX"
mkdir -p /sdin
mount ${device}2 /sdin
rm -f /sdin/etc/NetworkManager/system-connections/*
echo "" >/sdin/root/.ssh/known_hosts
echo "" >/sdin/home/archivista/.ssh/known_hosts
rm -f /sdin/home/archivista/.local/share/calls/*
rm -f /sdin/root/.local/share/calls/*
rm /sdin/var/log/Xorg*
echo "" >/sdin/root/.ash_history
echo "" >/sdin/home/archivista/.ash_history
umount /sdin
zerofree ${device}2
dd if=$device of=$1 bs=512 count=14880767
gzip $1
```

This creates a new master image. Again, /dev/sdX must be replaced with the correct letter. It is also important that the profiles for WLAN, phone calls and SMS are deleted. If you don’t want this, you can comment out the corresponding commands.



Developing with pmbootstrap

If you want to customize AVMultiPhone more profoundly, you will sooner or later resort to the development environment of postmarektOS. These are Python3 scripts. Unfortunately the installation command ‘pip3 install pmbootstrap’ does not always work. Therefore a small script was created for AVMultimedia:

```
#!/bin/bash

cp -f sudoers /etc
apt-get update
apt-get -y install python3-pip
apt-get -y install git-core
apt-get -y install binfmt-support
pip3 install pmbootstrap
export PATH=/home/archivista/.local/pmbootstrap/bin:$PATH
```

After the environment is available, you can start with **pmbootstrap init.** If everything works, you will have to answer a few questions, if not, error messages will appear. If the process is successful, an image can then be created:

`pmbootstrap install --sdcard=/dev/sdX`

Here too, the correct letter must be used. Other commands that are useful in the development environment:

pmbootstrap aportgen –fork-alpine xxx Create local copy for customizations
pmbootstrap kconfig edit linux-postmarketos-allwinner modify kernel file
pmbootstrap checksum linux-postmarketos-allwinner perform test

pmbootstrap build linux-postmarketos-allwinner –force rebuild kernel

The options **checksum** and **build** can be applied to any package, the packages themselves are managed by the file ‘APKBUILD’. In the kernel this file is located in the subfolder of the main pmOS folder:

`cache_git/pmaports/main/linux-postmarektos-allwinner`

There may also be copies (for forked and new packages) in the temp folder:

`~/pmOS/cache_git/pmaports/temp/xxx`

If a new package should always be created, the release number in APKBUILD should be increased. If the ‘build’ command is successful, the package can be obtained from the order below:

`~/pmos/packages/edge/aarch64`

If you need help, you can try your luck in the IRC channel #postmarketos. A web-based access is possible via **https://freenode.net** (without registration). The channel can also be accessed via **https://riot.im** (with registration). It is important to know that assistance is always voluntary, i.e. there is no guarantee of support.



Participate and win

Currently, AVMultiPhone is missing two functions that would make working much more comfortable. That’s why 500 euros each in premiums are being suspended here to maintain the two functions.

Awakening from deep sleep due to call

Calls from deep sleep are currently possible without problems from Mobian and UBPorts. With AVMultiPhone or pmOS this is currently not possible. According to a UBPorts developer the activation is done with the following two commands:

```
echo "AT+QDAI=1,0,0,2,0,1,1,1" | atinout - /dev/EG25.AT -
echo 'AT+QCFG="risignaltype", "physical"' | atinout -
/dev/EG25.AT -
```

These commands are also working under AVMultiPhone, ‘OK’ is returned. Telephone calls are also accepted during deep sleep, but the deep sleep is not successfully ended (i.e. the call cannot be noticed). Instead, AVMultiPhone remains in deep sleep until the power key is pressed and then rings briefly (the call therefore arrives in deep sleep, there is simply no way to answer the call directly from deep sleep at the moment.

For the premium either an AVMultiPhone image has to be delivered, where the waking up from deep sleep works in battery mode, or it has to be reproducibly documented how this can be retrofitted.

Commands to access camera modes

The main camera is realized via the kernel module ‘ov5640’. The following modes are documented in the sources:

```
OV5640_MODE_QVGA_176_144,
OV5640_MODE_QVGA_320_240,
OV5640_MODE_VGA_640_480,
OV5640_MODE_NTSC_720_480,
OV5640_MODE_PAL_720_576,
OV5640_MODE_XGA_1024_768,
OV5640_MODE_720P_1280_720,
OV5640_MODE_1080P_1920_1080,
OV5640_MODE_QSXGA_2592_1944,
```

These resolutions must be available as both image and video at 15, 30 and 60 frames per second. Of course only if the resolution allows it (e.g. the ov5640 sensor can only record HD with 1920×1080 and 30 fps).

Again, either an image can be made available with AVMultiPhone or it must be demonstrated with comprehensible steps how the functionality can be realized in AVMultiPhone.

Alternatively, offers can be submitted for the above or other jobs. Offers sent to **webmaster@archivista.ch** will be processed with pleasure in a friendly and speedy manner.

Update from June 30, 2020: *The award for the camera response has been awarded.*

All necessary information can be found at ***https://blog.brixit.nl/camera-on-the-pinephone.*** *Many thanks to the developer Martijn Braam. Further work for a graphical user interface for the camera is welcome, further payment cannot be guaranteed but will be gladly examined.*

Update from July 2, 2020: *With* ***postav_200701final.img.gz*** *calls are now accepted from sleep mode, but the runtime is "only" about 10 hours. Up to 20 hours are currently only available without waking up from deep sleep, for this there is* ***postav_200701deeptest.img.gz.*** *In other words, the problem with waking up is currently that the 'deep' mode is not working (reliably). The "prize money" for the deep mode remains open.*

Update from December 28, 2020: *Unfortunately, pmOS (and thus currently also AVMultiPhone) still has problems with the deep sleep mode. And because the developers of pmOS even in December 2020 neither know what the problem is nor have a schedule for fixing the problems, the decision had to be made that AVMultiPhone is currently no longer being developed.*