

Your Rubix model

It is important to identify your Rubix model. Currently, Rubix version is v1.1 base A10 chip, it has 1GB RAM and 4GB NAND.

Connecting the parts

You can connect your HDMI monitor/TV using an HDMI cable to Rubix. Note that if your HDMI monitor/TV cannot display 1080p, then it might not be able to display properly on the screen. The Rubix display system default setting is 1080p resolution.

You can directly connect a keyboard and mouse to the two USB port on Rubix or connect a USB hub and then attach a keyboard and mouse. Note that if you connect several devices, such as USB hard drive with 1TB+ to the Rubix, then you will need to provide more external USB power or AC-DC power (for example, 500mA will probably be not enough).

You can power your Rubix from either

- an AC-DC Adapter(recommended)
- a USB to mini-USB cable

You can plug the AC-DC adapter (12V/1.5A+) to power bar or a cable to USB port with on your computer, and the other end (either DC or mini-USB) to the appropriate port on the Rubix. It is very important to power your Rubix stably, an AC-DC adapter is recommended.

First startup

Note if you first startup the Rubix, DO NOT attach a micro SD card with any bootable OS image, it may damage Rubix system.

Now you can power the Rubix using either an AC-DC adapter or a powered USB to mini-USB cable. The Rubix should power on - the white *POWER LED* should light up.

It is about 35 seconds, and then it startup a desktop from the NAND flash memory. The Rubix has 4GB NAND, which pre-load Linaro with essential applications, such as Arduino IDE.

Power Off and On

Once you connect the Rubix to the power, it will automatically boot up and auto login “root” account (default password is “password”).

It is recommended to reboot/power off the Rubix using the facility of the operating system. You can reboot/power off either from menu bar on desktop (recommended) or run shell command in terminal (if you are a Linux hobbyist).

#reboot/poweroff

Note that you CAN NOT switch user from desktop, by default Rubix auto login “root” account with administrator privileges. You can add another account when you need.

It is NOT recommended to forcibly shutdown Rubix by keeping long pressed the *Power* button. It may damage your Rubix file system.

Note if your Rubix power off exceptionally, you can power on Rubix to check boot up correctly, if so then reboot again to double check system running correctly.

Resources and Tutorials

Rubix is based on A10 chip, so it can use more stable and common open source platform both Linux kernel and ARM embeded system.

You can get a lot of tutorials from Allwinner A10 [Sunxi platform community](#), and more resources also get from the [open source Allwinner SoCs](#).

NOTE that any tutorials have no stable or unique answer, enjoy any hack.

Advanced Support

We are glad to transfer any best practices for you if advanced technical support needed. Note that advanced support is FREE for all our customers.

Some of best practices we try to provide may NOT fit for others boards which also based on Allwinner A10 chip.

Don't hesitate email us when you want to customize your Rubix, such as microSD card image and instruction.

Best Practices for Rubix

How to build a new bootable image for microSD card

We will get there through these steps: download original images, burn images into microSD card, transfer root file system into microSD card, reboot Rubix from microSD card, and build your favorite Linux is optional.

Follow these detailed steps.

Open a browser from your desktop or open a terminal from Rubix Desktop (you can use shortcut keys Ctrl + Alt + T to open it).

Download Rubix original bootable images for microSD card from these links, or execute the command in terminal.

Example:

```
wget -r https://www.dropbox.com/s/5925v3zfbw1cgm8/mmcuboot.img?dl=1 -o /opt/mmcuboot.img
```

```
wget -r https://www.dropbox.com/s/x80qk3cdwhsbbtv/mmcbootloader.img?dl=1 -o /opt/mmcbootloader.img
```

```
wget -r https://www.dropbox.com/s/vflv40cz0k2bpp/mmcrubix.tar.gz?dl=1 -o /opt/mmcrubix.tar.gz
```

Insert your new 4GB+ microSD card into Rubix and execute the command

Example:

```
#umount all partitions of new microSD card, ls /dev/mmcblk* get info
```

```
#ATTENTION any /dev/nand* partitions are Rubix internal partitions, DO NOT damage them.
```

```
cd /opt
```

```
umount /dev/mmcblk0p1 /dev/mmcblk0p2
```

```
dd of=/dev/mmcblk0 if=mmcuboot.img bs=1M count=1
```

```
#refresh new partitions info of mmcblk0
```

```
sfdisk -R /dev/mmcblk0
```

```
umount /dev/mmcblk0p1 /dev/mmcblk0p2
```

```
dd of=/dev/mmcblk0p1 if=mmcbootloader.img bs=1M count=16
```

```
umount /dev/mmcblk0p1 /dev/mmcblk0p2
```

```
mkfs.ext4 -L rootfs /dev/mmcblk0p2
```

#Convenient for next time, we put above command lines together.

```
cd /opt;umount /dev/mmcblk0p1 /dev/mmcblk0p2;dd of=/dev/mmcblk0  
if=mmcuboot.img bs=1M count=1;sfdisk -R /dev/mmcblk0;dd  
of=/dev/mmcblk0p1 if=mmcbootloader.img bs=1M count=16;umount  
/dev/mmcblk0p1 /dev/mmcblk0p2;mkfs.ext4 -L rootfs /dev/mmcblk0p2;
```

Now you get a new bootable image for microSD card, next step transfer our Rubix big root file system into mmcblk0p2 partition, it will be about ten minutes, enjoy your favorite cup of coffee during executing the command

Example:

```
rsync -aAxXH --numeric-ids /* /media/rootfs/  
--exclude={"/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/lost+found/*}  
&>/dev/null;sed -i "s/nandc/mmcblk0p2/g" /media/rootfs/etc/fstab
```

#or

```
umount /dev/mmcblk0p2;mount /dev/mmcblk0p2 /mnt;cd /mnt;tar -xpf  
/opt/mmcrubix.tar.gz &>/dev/null;sed -i "s/nandc/mmcblk0p2/g" /mnt/etc/fstab
```

If correctly finish the command, you will get a bootable Linaro from microSD card. If encounter errors or stuck, you could redo the command.

Now you can try reboot Rubix from your microSD card (DO NOT plug out card).

Note that booting Rubix from microSD card is slower than NAND, it will take 1-3 minutes. If it is still not boot into Rubix desktop over 5 minutes, power off and boot again, if it cannot boot into Rubix desktop more times, don't worry, your internal Rubix desktop still be in NAND, you need rebuild microSD card follow above instruction or email us get help.

How to restore original Rubix root file system for NAND

When you finish a new bootable original image for microSD card, you can use it to restore original Rubix root file system for NAND if your Rubix really encounter errors or bootup incorrect all the time.

ATTENTION that any project development files or your important data should be backup into your external storage before restoring root file system.

Rubix is able to run as a Linux server, it means any common Linux tech will be used on Rubix root file system of NAND or microSD card. Following the same tech, you can restore original Rubix root file system for NAND.

Power off Rubix and Insert your microSD card with original image into Rubix, then power on again. After showing desktop a few seconds then execute the command in Linux terminal.

Example:

```
umount /dev/nandc
```

```
mkfs.ext4 -L rootfs /dev/nandc
```

```
umount /dev/nandc;mount /dev/nandc /mnt
```

Now you get a new clean file system for NAND, next step transfer our Rubix big root file system into nandc partition, it will be about ten minutes, enjoy your favorite cup of coffee again during executing the command

Example:

```
rsync -aAxXH --numeric-ids /* /mnt/  
--exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/run/*,/mnt/*,/media/*,/lost+found/*}  
&>/dev/null;sed -i "s/mmcblk0p2/nandc/g" /mnt/etc/fstab
```

#or tar original rootfs if you prepare original image file into /opt

```
cd /mnt;tar -xpf /opt/mmcrubix.tar.gz &>/dev/null;sed -i "s/mmcblk0p2/nandc/g"  
/mnt/etc/fstab
```

If correctly finish the command, you will get a bootable Linaro from NAND. If encounter errors or stuck, you could redo the command.

Now you can power off Rubix and plug out microSD card, power on again and try boot up Rubix from NAND.

How to build your favorite Linux Image for microSD card

When you get a new bootable image for microSD card, you still can use the rsync command to transfer your favorite Linux into another microSD card. Note that your favorite Linux root file system must be built for ARM and with Rubix kernel (/lib/modules/3.4.61.sun4i+).