

**Suzie Linux** <https://suzielinux.com/>

Suzie Linux was named in memory of my adorable Maine Coon cat Suzie.

## **Archlinux for YY3568 boards documentation**

This documentation is release under the GPL Licence 2.0

<https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

Author	Date	Project	Revisions
Michel Catudal	2025-06-08	<b>YY3568 Archlinux Creation</b>	<b>1</b>

# REVISION TRACKING SHEET

[illegible]

<h1>Table of Content</h1>
---------------------------

.....1

**Suzie Linux <https://suzielinux.com/>.....1**

**Suzie Linux was named in memory of my adorable Maine Coon cat Suzie.....1**

**1. Hardware.....4**

1.1. OVERVIEW OF THE YY3568 BOARD.....4

**2. Gentoo applications required for chroot.....6**

2.1. GENTOO APPLICATIONS REQUIRED.....6

2.2. APPLICATIONS REQUIRED FOR CHROOT ON MAC VMWARE FUSION DEBIAN.....7

**3. Archlinux Root File System.....7**

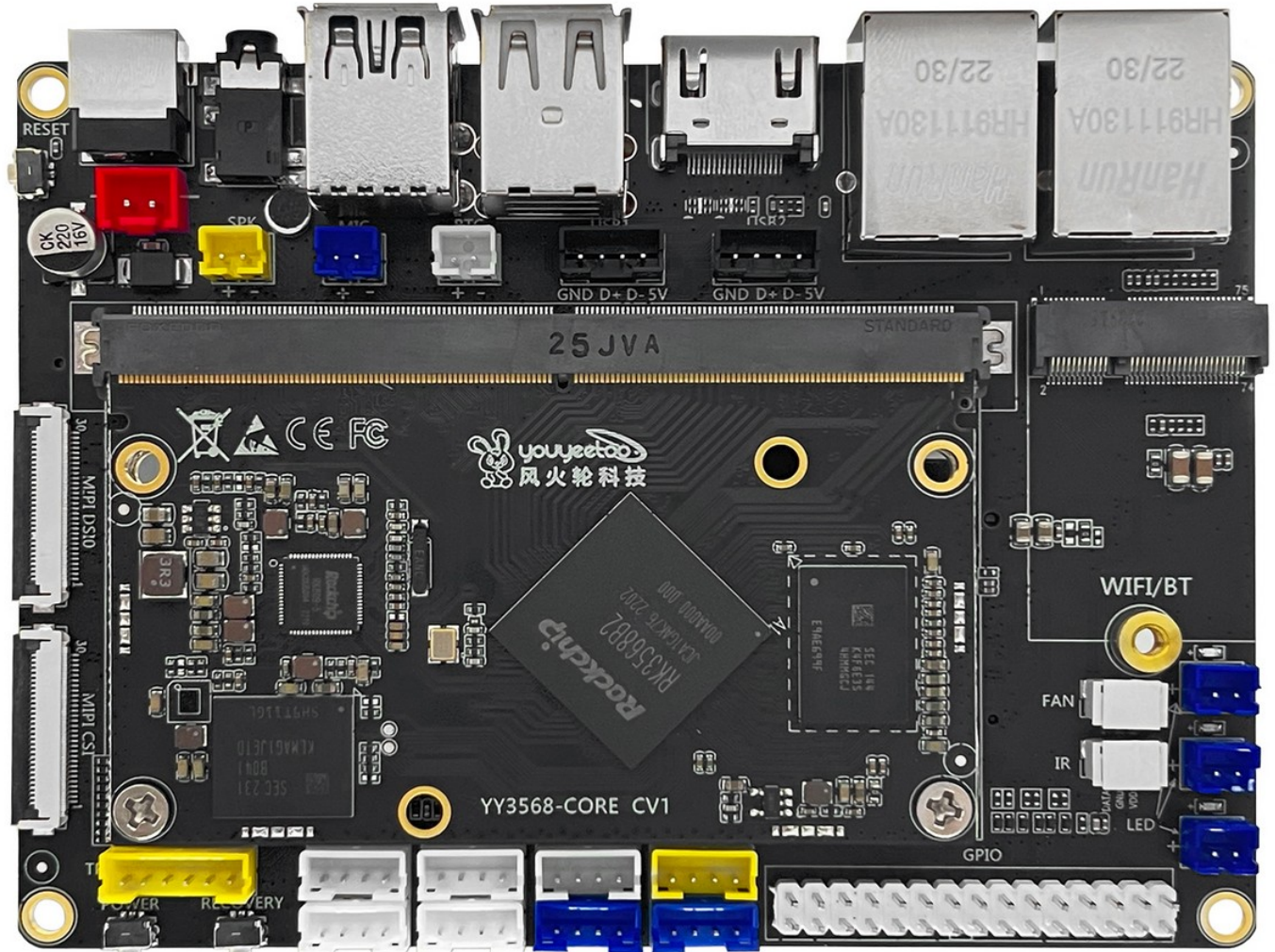
3.1. CREATE A ROOT FILE SYSTEM.....7

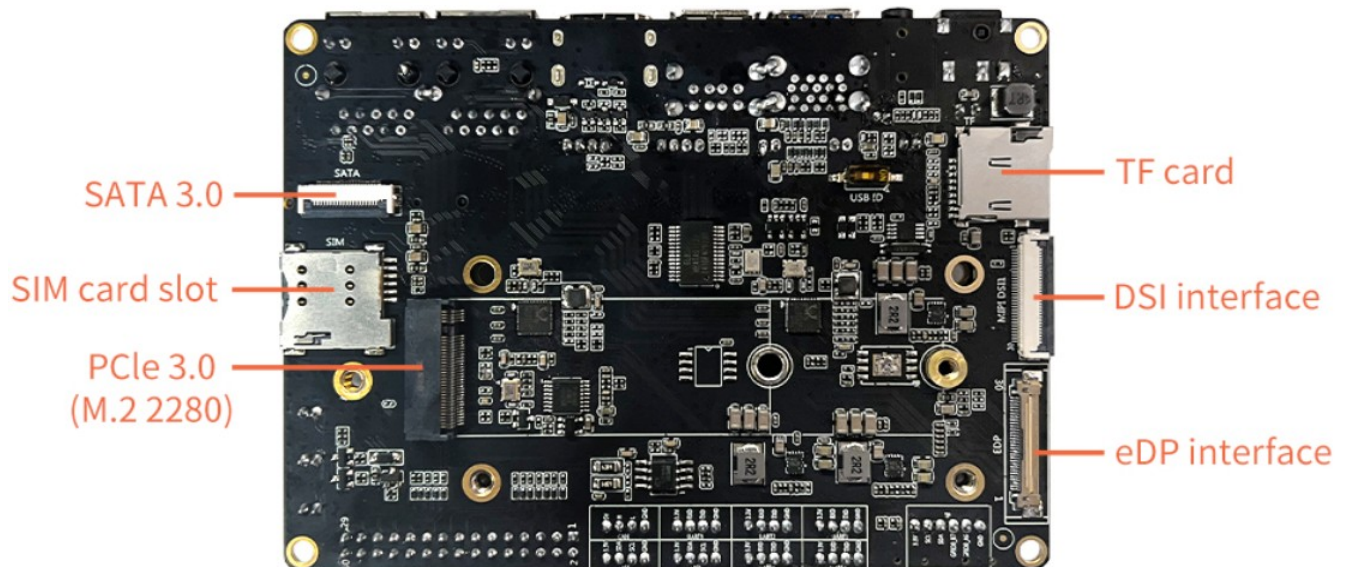
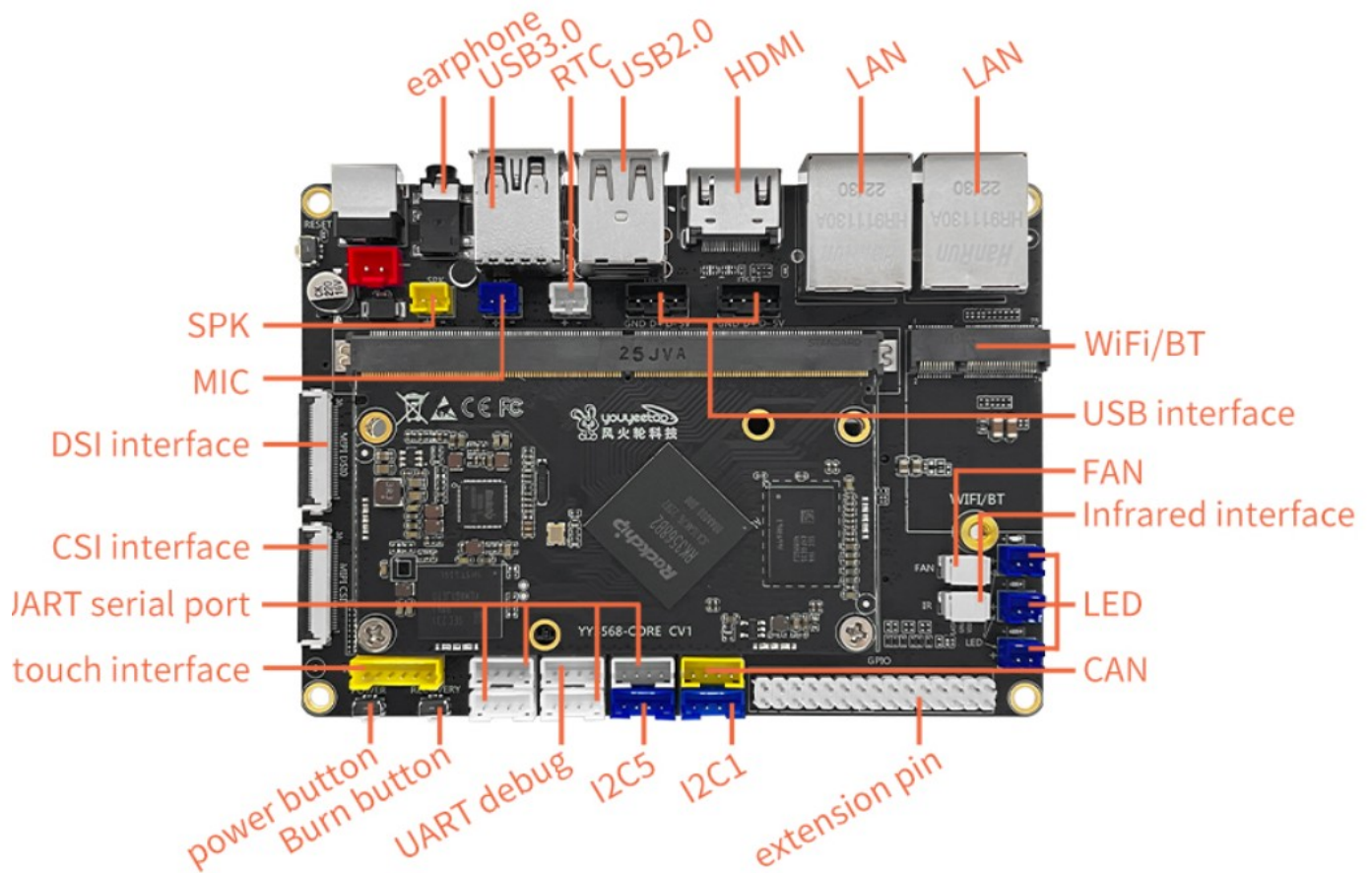
3.2. CHROOT INTO ARCHLINUX ROOTFS.....8

**4. Create archlinux micro SD boot disk.....9**

## 1. Hardware

### 1.1. Overview of the YY3568 board





## 2. Gentoo applications required for chroot

```
cd ~
mkdir yy3568
cd yy3568
export work_directory=$(pwd)
Get misc files needed for the bootloader and rootfs
git clone https://github.com/SuzieLinux/YY3568.git files
```

### 2.1. Gentoo applications required

```
emerge --ask dev-python/cryptography
emerge --ask dev-python/pyelftools
emerge --ask dev-util/yamllint
emerge --ask dev-libs/libyaml
emerge --ask dev-python/jsonschema
emerge --ask sys-block/bmap-tools
emerge --ask sys-fs/genimage
emerge --ask sys-fs/mtools
emerge --ask gnutls
emerge --ask flex
emerge --ask sys-devel/bc
emerge --ask bison
emerge --ask swig
emerge --ask sys-fs/dosfstools
cd /usr/bin
ln -s mkfs.vfat mkdosfs
emerge --ask sys-apps/arch-chroot
```

In order to chroot on a arm64 rootfs a few things have to be done.  
 First you need to make sure that the kernel supports it and emerge needed support  
 The build system's kernel must support miscellaneous binary formats.  
 This can be enabled with CONFIG\_BINFMT\_MISC=m  
 or CONFIG\_BINFMT\_MISC=y in the the kernel's **.config** file.

A system restart is required after building this module before it can be used.

#### **Enable CONFIG\_BINFMT\_MISC**

```
Executable file formats --->
<*> Kernel support for MISC binaries
```

USE=static-user needs to be set

Add this to /etc/portage/package.use/qemu :

```
# Enable static-user and add the arm64 and other targets
app-emulation/qemu static-user QEMU_SOFTMMU_TARGETS: * QEMU_USER_TARGETS: *
# required by app-emulation/qemu::gentoo[static,static-user]
# required by qemu (argument)
dev-libs/glib static-libs
# required by app-emulation/qemu::gentoo[-static,static-user]
# required by qemu (argument)
sys-libs/zlib static-libs
# required by app-emulation/qemu::gentoo[-static,static-user,xattr]
```

```
# required by qemu (argument)
sys-apps/attr static-libs
# required by dev-libs/glib::gentoo
# required by app-emulation/qemu::gentoo[-static,static-user]
# required by qemu (argument)
dev-libs/libpcr2 static-libs
```

```
emerge --ask app-emulation/qemu --update --newuse --deep
```

## 2.2. Applications required for chroot on mac vmware fusion debian

```
sudo apt upgrade
sudo apt install build-essential git vim
sudo apt install gfortran gpc
sudo apt install debhelper fakeroot
sudo apt install python3-cryptography
sudo apt install python3-pyelftools
sudo apt install yamllint
sudo apt install libyaml
sudo apt install libyaml-dev
sudo apt install python3-pyelftools
sudo apt install python3-jjsonschema
sudo apt install python-jjsonschema
sudo apt install bmap-tools
sudo apt install genimage
sudo apt install dosfstools
sudo apt install mtools
sudo apt install gnutls-dev
sudo apt install flex
sudo apt install bc
sudo apt install bison
sudo apt install swig
sudo apt install arch-chroot-scripts
```

## 3. Archlinux Root File System

```
export rootfs_dir=$work_directory/archlinux_rootfs
cd $rootfs_dir
```

We download the latest rootfs available

```
wget http://os.archlinuxarm.org/os/ArchLinuxARM-aarch64-latest.tar.gz
```

### 3.1. Create a root file System

Build on gentoo is very slow so I created it on the mac using vmware fusion  
That could probably be done on a RPi5 or Beaglebone AI 64

```
su
mkdir -p $rootfs_dir
tar xfp ArchLinuxARM-aarch64-latest.tar.gz -C $rootfs_dir
sync
cp /usr/bin/qemu-aarch64 $rootfs_dir/usr/bin
cd $rootfs_dir/etc
cp $work_directory/archlinux/files/etc/locale.gen ./
```



```
cp /etc/resolv.conf ./
cp $work_directory/archlinux/files/etc/bash.bashrc ./
cp $work_directory/archlinux/files/etc/fstab ./
cp $work_directory/archlinux/files/etc/hostname ./
sed 's/CheckSpace/#CheckSpace/g' pacman.conf
cd $rootfs_dir/boot
cp -R $work_directory/archlinux/files/misc/boot/extlinux ./
cd ..
cp -R $work_directory/archlinux/files/misc/arch_packages.tar.gz ./
```

### 3.2. chroot into archlinux rootfs

```
cd $work_directory
It is assumed here that you are still under root
```

```
arch-chroot $rootfs_dir
source /etc/profile
```

This is needed when chroot on gentoo but is not needed on the mac  
It actually crashes chroot if you do  
`export PS1="(chroot) $PS1"`

We need a user for later login thru ssh  
`userdel -r alarm`  
`useradd -m suzie`

```
pacman-key --init
pacman-key --populate archlinuxarm
locale-gen
```

Here I create simple passwords, after we boot the micro sd we can change them to more secured password. For all our settings in chroot this approach makes work simple. In both case it will ask to confirm the password.

For the root password : `passwd`  
For the suzie user password : `passwd suzie`

For the time eastern time zone  
`ln -sf /usr/share/zoneinfo/America/Detroit /etc/localtime`

```
pacman -Syuu
pacman -S python-setuptools tk sqlite dtc vboot-utils uboot-tools bc
pacman -S inetutils docbook-xsl smlto dhcpd pacman-contrib glibc
systemctl enable dhcpd
pacman -S base-devel perl git subversion wget vim openssh
systemctl enable sshd
pacman -S dtc python python-setuptools tk sqlite docbook-xsl xmlto
pacman -S uboot-utils uboot-tools bc inetutils
```

```
cd /home/suzie
mv /arch_packages.tar.gz ./
chown suzie:suzie arch_packages.tar.gz
su suzie
tar xvf arch_packages.tar.gz
cd arch_packages
```



```
cd joe
makepkg
cd ../linux-yy3568
makepkg
exit
```

```
cd arch_packages
cd joe
pacman -U joe-4.6-2-aarch64.pkg.tar.xz
cd ../linux-yy3568
pacman -U linux-yy3568-6.15.1-1-aarch64.pkg.tar.xz
pacman -U linux-yy3568-headers-6.15.1-1-aarch64.pkg.tar.xz
```

Setup some links to simulate the cpm-80 wordstar editor

```
cd /usr/bin
ln -s joe ws
cd /etc/joe
cp jstarrc wsrc
```

To remove the annoying wordwrap bug delete all mentions of wordwrap in wsrc  
It gets to be a pain when you update a script and it cuts a line and you didn't notice  
Your script has no chance of working with this ridiculous behavior of the editor.  
exit

```
cd $work_directory/archlinux_rootfs
NOW=$(date +"%Y%m%d%H%M")
sudo tar cvfJ $work_directory/archlinux-yy3568-rootfs-$NOW.xz *
```

#### 4. Create archlinux micro SD boot disk

If chroot was not done on gentoo it needs to be copied to the gentoo disk

```
cd ~/yy3568
export work_directory=$(pwd)
```

The needed files should be on \$work\_directory/archlinux/files

```
sudo $work_directory/archlinux/files/scripts/mk_archlinux_rootfs.sh
```

We create the SD boot disk using a working u-boot image.

```
uboot_image=$work_directory/archlinux/files/misc/boot/uboot.img
archlinux_rootfs=$work_directory/input/rootfs.ext4
```

Change sdd to whatever you micro SD is on

```
su
dd if=$uboot_image of=/dev/sdd bs=512
sync
dd if=$archlinux_rootfs of=/dev/sdd1 status=progress iflag=direct oflag=direct bs=4M
sync
partprobe /dev/sdd
fsck.ext4 -f /dev/sdd1
resize2fs /dev/sdd1
```