Suzie Linux https://suzielinux.com/

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

Open Suse 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	2025-06-12	**YY3568 Open Suse Creation**	**1**

REVISION TRACKING SHEET

Rev	Name	Date	Comment
1	Michel Catudal	2025-06-12	

Table of Content

1. Hardware

1.1. Overview of the YY3568 board
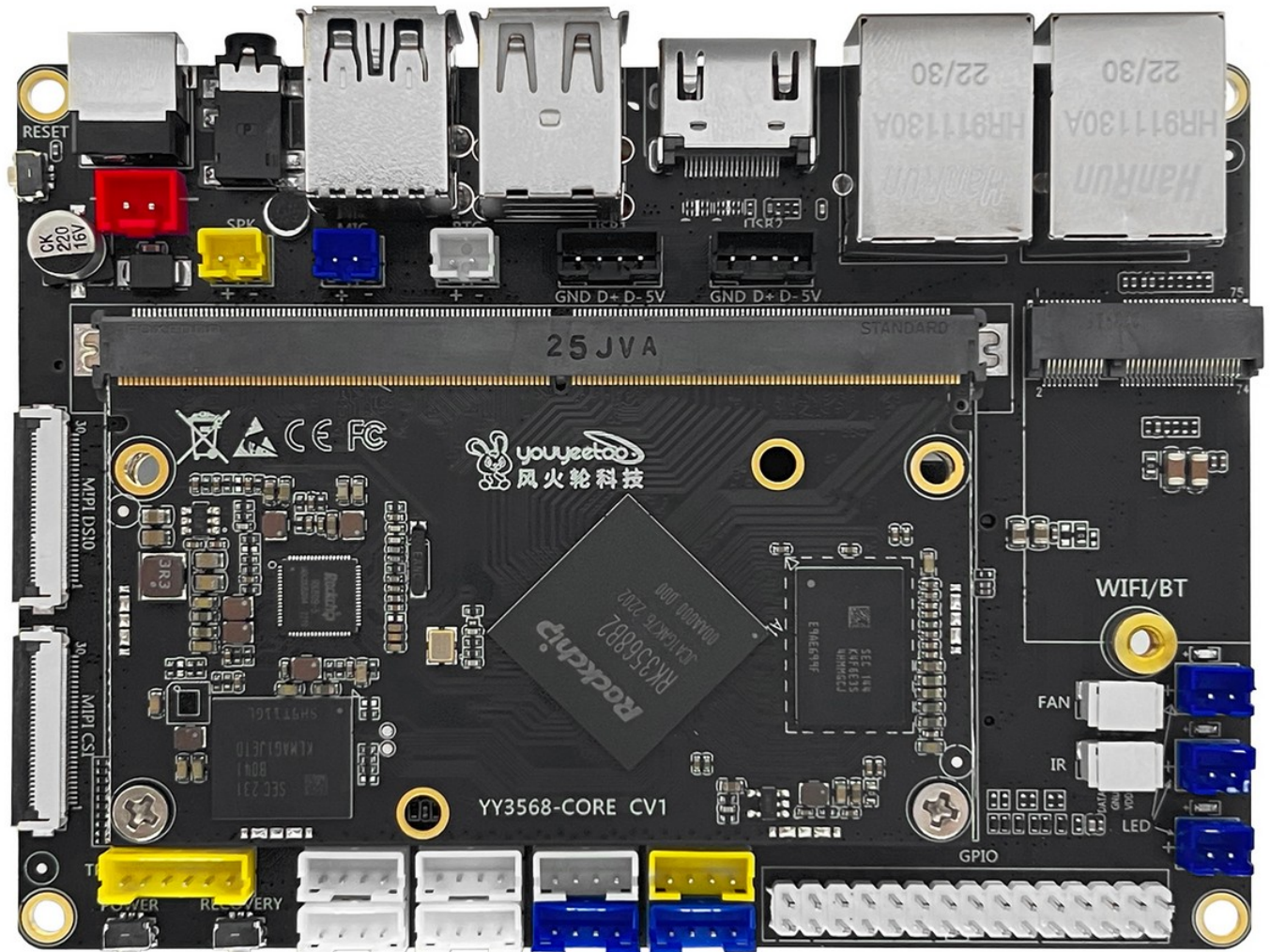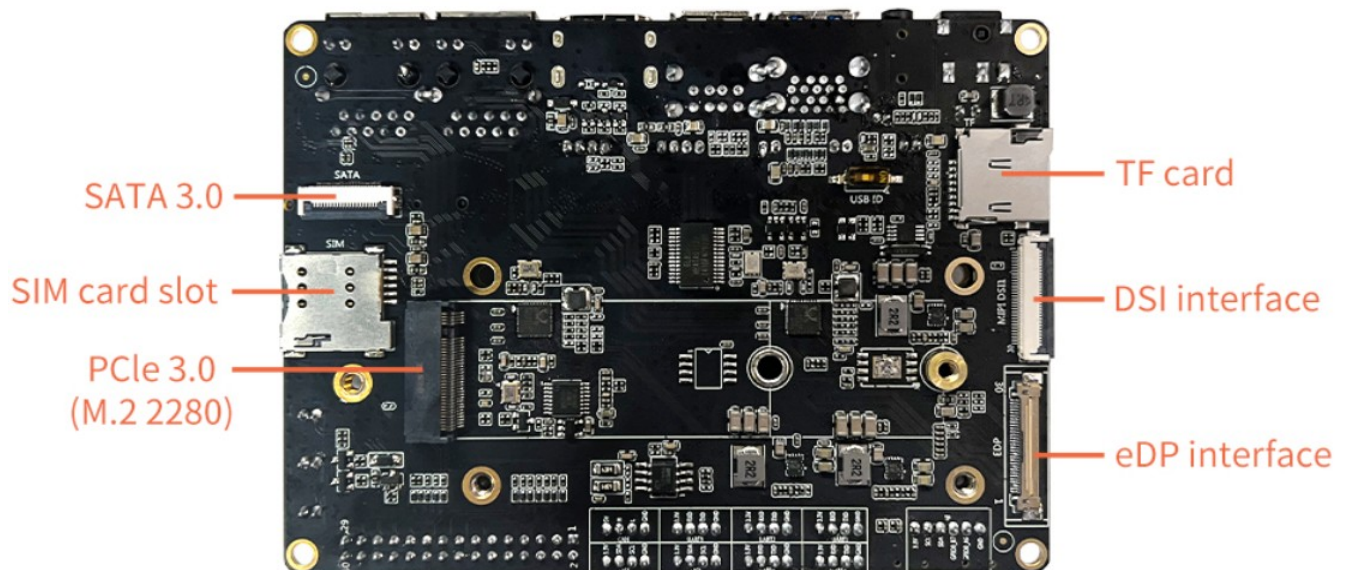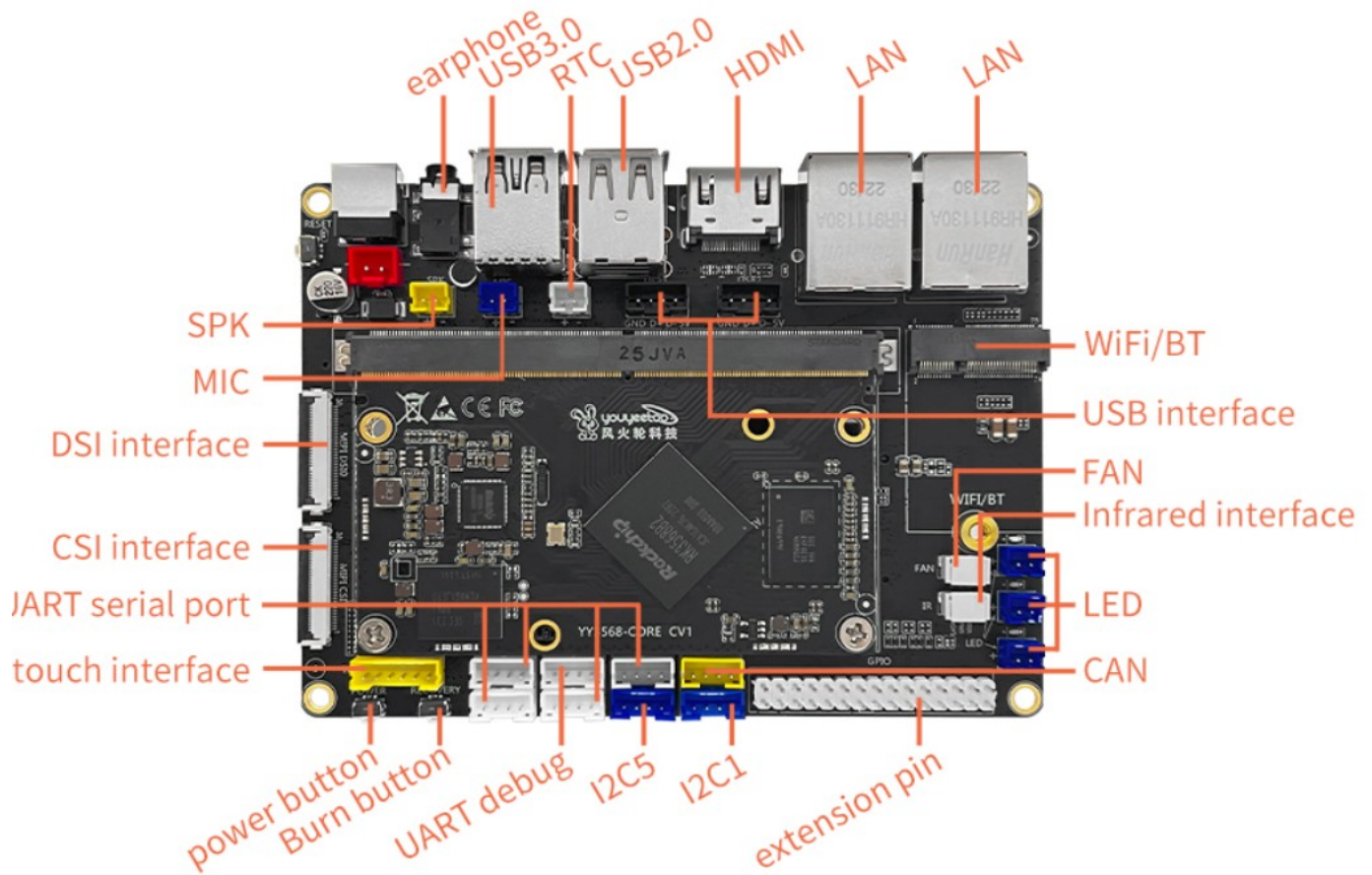
2. <u>Gentoo applications required for chroot</u>

```
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/libpcre2 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-jsonschema
sudo apt install python-jsonschema
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. Open Suse Root File System

```
export rootfs_dir=$work_directory/opensuse_rootfs
export suse_download=https://mirrorcache.opensuse.org/download
export suse_files_dir=$suse_download/ports/armv7hl/factory/appliances
cd $rootfs_dir
```

We download the latest rootfs available

```
wget $suse_files_dir/openSUSE-Tumbleweed-ARM-LXQT.aarch64-rootfs.aarch64.tar.xz
```

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 done on a RPI5 or Beaglebone AI 64
export misc_files=$work_directory/opensuse/files

```
su
mkdir -p $rootfs_dir
tar xfvp opensuseARM-aarch64-latest.tar.gz -C $rootfs_dir
sync
cp /usr/bin/qemu-aarch64 $rootfs_dir/usr/bin
```

```
cd $rootfs_dir/etc
cp /etc/resolv.conf ./
cp $misc_files/etc/bash.bashrc ./
cp $misc_files/etc/fstab ./
cp $misc_files/etc/hostname ./
cp $misc_files/0001-Added-support-for-YY3568.patch $rootfs_dir/usr/src
cp $misc_files/ config-6.15.0-1-catu $rootfs_dir/usr/src
cd $rootfs_dir/boot
cp -R $misc_files/misc/boot/extlinux ./
```

3.2. chroot into opensuse 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
useradd -m suzie
zypper install kernel-source
zypper remove selinux-tools selinux-policy
zypper install rpm-build
zypper install net-tools
zypper install joe

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.

cd /usr/src/linux-6.15.0-1
make mrproper
cp ../config-6.15.0-1-catu .config
patch -p1 <../0001-Added-support-for-YY3568.patch
make vmlinux
make modules
make dtbs
make modules_install
make install
make dtbs_install
cp arch/arm64/boot/Image /boot/vmlinuz-6.15.0-1-catu
cp System.map /boot/System.map-6.15.0-1-catu
cp .config /boot/config-6.15.0-1-catu
exit
```

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

4. Create opensuse 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/opensuse/files

```
sudo $work_directory/opensuse/files/scripts/mk_opensuse_rootfs.sh
```

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

```
uboot_image=$work_directory/opensuse/files/misc/boot/uboot.img
opensuse_rootfs=$work_directory/input/rootfs.ext4
```

Change sdd to whatever you micro SD is on

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